Universidad Carlos III de Madrid
Escuela Politécnica Superior

Grado en Ingeniería en Informática

Trabajo de Fin de Grado

Análisis, diseño e implantación de un sistema de monitorización de aulas departamentales

Estudiante: Ivar Giovanni Quiroz Rocco
Tutor: Alejando Calderón Mateos

Índice de contenidos

Índice de Ilustraciones

Índice de tablas

Agradecimientos

Al lector, fuera de todo ámbito profesional:

No puedo evitar ser emotivo en este apartado. Han sido muchas las vivencias que me han llevado a este documento, muchas de ellas son algo duras, pero el total de la experiencia suma positivo. Este apartado va dedicado a aquellos que me conocen y saben que soy una persona emocional.

A todos vosotros:

Ha sido difícil llegar hasta aquí, ser independiente, vivir solo, cambiarme de país, dejar a mi familia, afrontar la vida lejos de aquellos que amo, asistir a clases por la mañana, trabajar por las tardes y fines de semana… Ha sido muy difícil, pero no ha sido en vano, he conseguido terminar la carrera, he conocido a gente maravillosa y he encontrado a alguien muy especial.

Por esto y muchas cosas más, quiero dar las gracias primero a Carmen Figueras, por no dejar que me rindiese, por no permitir que me faltase nada, por apoyarme siempre en todas mis decisiones y por ayudarme a conseguir este sueño, este gran y valioso sueño, que nos va a permitir realizar todos nuestros proyectos futuros. Cariño, te amo, estoy infinitamente agradecido de todo lo que haces por mí y como te he dicho en mil ocasiones, esto no habría sido posible si no hubieses estado a mi lado todo este tiempo. Este documento va dedicado a ti y a todo tu esfuerzo por hacerme feliz.

Quiero dar las gracias a mi familia, por haberme educado de una forma especial, por enseñarme el *semper fidelis*, el *carpe diem* y por enseñarme a no rendirme jamás. Ha habido muchos momentos en los cuáles podría haber terminado en otro rumbo, pero ha sido gracias a ellos que he sabido como alzar el vuelo y encontrar el mío propio.

Quiero creer que mi abuela puede leer este documento desde el infinito, quiero creer que en alguna parte de universo sigue existiendo ese calor maternal que me diste todo este tiempo. Sé que he sacrificado el pasar a tu lado tus últimos años por cumplir este gran sueño, pero quiero que sepas que he rendido honor a tu memoria y a la educación que me diste... he cumplido la promesa que te hice, he terminado la carrera. Te amo y quiero que sepas que no voy a olvidarte nunca.

También quiero agradecer a mis amigos, compañeros y profesores de la Universidad Carlos III de Madrid por no haber permitido que mis ánimos decayesen nunca, gracias a su apoyo y cariño conseguí ser feliz todos estos cinco años. Gracias al laboratorio del Departamento de Informática y a todos sus integrantes, a la real sociedad de pingponeros informáticos, gracias por los ágapes, los viernes por la tarde, los eternos buenos momentos, gracias por abrirnos las puertas y confiar en nosotros, éste es el fruto de vuestro apoyo incondicional.

Todas las lágrimas, sudor y sangre las he compartido con vosotros estos últimos años, quiero que sepáis que, aunque vuelva a mi país de origen **siempre os llevare en mi corazón**.

Por último, para todas aquellas personas que lean este documento: siempre que estéis pasando por momentos de flaqueza y debilidad… **¡ANIMO!, las grandes batallas de la vida siempre se ganan, pero para ello…**

I.Q.

Resumen

Este trabajo de fin de grado surge de la necesidad de poder **controlar lo que ocurre con las aulas informáticas de la Universidad Carlos III de Madrid**, ya que el sistema implantado actualmente permite identificar sucesos puntuales por medio de gráficos históricos, pero no permite comprender el motivo por el cual ocurren dichos sucesos.

Por medio de la investigación realizada y entregada en este documento se **pretende ofrecer una alternativa eficiente y escalable a las necesidades de la Universidad** y con ello, poder mejorar el manejo y uso de los recursos de las aulas informáticas de la misma, obteniendo información crítica para identificar los motivos de las caídas de servicios (intentos de ataque, fallas de *kernel*, problemas con librerías, etc.).

1. Introducción

En este apartado se definirá cual será **la visión general de este documento**. Para ello, se explicará a rasgos generales cuáles son las necesidades que llevan a este proyecto y los objetivos a cumplir. También detallaremos cada uno de los apartados y el contenido de cada uno.

1.1. Motivación

Las aulas informáticas del laboratorio del Departamento de Informática (4.0.F16 y 4.0.F18) de la Escuela Politécnica Superior son usadas constantemente por estudiantes de distintas disciplinas. Están abiertas de lunes a viernes desde las 9:00hrs hasta las 21:00hrs, tienen instalados dos sistemas operativos (*Windows* 7 y *Debian* 8) y normalmente se usan para realizar clases y prácticas de las asignaturas de las distintas carreras que se imparten en el campus.

En ciertos momentos del curso académico, **los ordenadores reciben un uso intensivo**. El Departamento de Informática cuenta actualmente con un sistema de monitorización llamado Munin (1), el cual muestra información gráfica casi en tiempo real sobre el uso de recursos (CPU, memoria, red, etc.).

Debido a que Munin dibuja los datos, pero no almacena el contenido de los *logs*, es imposible recuperar información avanzada sobre el uso actual de recursos ni menos sobre lo que ha pasado anteriormente. El sistema se encarga de dibujar la continuidad del uso de los recursos, **pero no almacena la información** relativa a los mismos **ni cuenta con un sistema que permita consultar información** sobre ellos, lo que lo convierte en un sistema limitado frente a las necesidades actuales del laboratorio.

Bajo este contexto, se me ha ofrecido realizar un estudio sobre cómo mejorar esta situación. El personal del laboratorio necesita una herramienta que permitiera consultar información sobre el uso de recursos, en cualquier momento y con la mayor cantidad de detalles importantes posibles.

1.2. Objetivos

En resumen, **se pretende implementar una solución que permita monitorizar** de forma avanzada **el uso de recursos de las aulas del laboratorio** del Departamento de Informática. Esto permitiría, por ejemplo, identificar las caídas de servicio de dichas aulas y el motivo por el cual ocurren. También es necesario identificar problemas indirectos (p.e.: ocasionados por el desgaste, continuidad de uso, etc.) que puedan estar afectando a corto o largo plazo a dichos recursos.

Basado en lo antes descrito, los objetivos son:

- La solución debe ser capaz de realizar las mismas tareas que el actual sistema de monitorización de la Universidad (Munin), las cuáles son: monitorizar el uso de recursos de todos los ordenadores de las aulas informáticas 4.0.F16 y 4.0.F18, por medio de una interfaz gráfica y del uso de archivos de *log*, teniendo un histórico que permita detectar caídas de sistemas.

- La solución debe permitir búsquedas de la información contenida en dichos *logs*, de tal forma que se pueda obtener información parcial o total de los mismos.

- La solución debe permitir la obtención de dicha información por medio de los recursos del laboratorio de informática, es decir, la solución debe comunicarse por medio de una red *LAN* y la información debe poder enviarse desde los ordenadores de las aulas hacia un *servidor*.

- La interfaz gráfica de la solución debe permitir hacer *querys* con distintos criterios y detalles. El resultado debe mostrarse de forma gráfica o poder obtenerse como un conjunto resultado de datos.

Adicionalmente a esto, se debe tomar en cuenta la situación económica actual de la Universidad. Los recursos económicos destinados a la solución deben ser lo más reducidos posibles, por lo tanto:

- La solución será preferentemente de licencia gratuita y *open source*.

- El mantenimiento de la solución debe ser nulo o poco costoso en horas hombre y en coste de equipos (implantación y mantenimiento).

1.3. Estructura del documento

En este apartado se enumerará los distintos capítulos que integran este documento junto con un pequeño resumen de sus contenidos:

- **Capítulo 1: introducción.** En este apartado se realizará una generalización de cómo será el proyecto, especificando el contexto en el cual se realizó, los objetivos que se pretende cumplir y cuál será la estructura del documento junto con los acrónimos utilizados en el mismo.

- **Capítulo 2: Estado del arte.** En este apartado se revisarán las características generales de las distintas herramientas que se plantean como una solución a las necesidades de la Universidad y una pequeña comparativa entre ellas.

- **Capítulo 3: Análisis.** En este apartado se revisará las opciones contempladas en el apartado anterior y se decidida cual es la más apta ser usada como solución.

También se describirá la metodología que se usara para el análisis del problema, se modelara de qué forma se usara la solución para satisfacer dicho problema y se especificarán los requisitos del sistema.

- **Capítulo 4: Diseño.** En este apartado se describirá a fondo cuáles son los distintos módulos que integran la solución, cómo interactúan entre ellos y como con esa interacción satisfacen los objetivos.

- **Capítulo 5: Implantación.** En este apartado explicaremos cual será el procedimiento integral mediante el cual se introducirá la solución al entorno real, es decir, el proceso de instalación, configuración y puesta en marcha de la aplicación.

- **Capítulo 6: Pruebas.** En este apartado se describirá cuáles son las pruebas que se aplican a la solución para poder validar y garantizar el buen funcionamiento de la misma.

- **Capítulo 7: Planificación y presupuesto.** A lo largo de este apartado explicaremos cual será la planificación que se seguirá y que costes conlleva la realización del proyecto. Se explicará de forma gráfica y detalladamente la programación estimada de las fases del proyecto.

- **Capítulo 8: Conclusiones y trabajos futuros.** En este apartado detallaremos las conclusiones del desarrollo del proyecto y que futuros puntos de ampliación existen. Todas aquellas ideas que permitan expandir el alcance del proyecto serán indicadas aquí.

- **Apéndice I: Manual de instalación y configuración.** En este apartado es el manual de instalación y configuración que podrá ser utilizado por los técnicos del laboratorio de informática o aquellas personas que deseen implantar el proyecto en un entorno similar.

- **Apéndice II: Manual de utilización.** En este apartado se mostrará el manual de utilización. Con él, los usuarios finales que estén a cargo de la solución podrán tener una guía de aprendizaje que les permitirá entender el funcionamiento de la misma.

- **Apéndice III: Detalle de la planificación, diagrama de Gantt.** En este apartado podremos visualizar los detalles de la planificación mostrada en el capítulo 7, podrá usarse para ver en grande e imprimir horizontalmente el diagrama de Gantt.

- **Bibliografía.** Finalmente, en este apartado se indicarán todas las referencias bibliográficas y fuentes que se han consultado en el desarrollo de este proyecto.

1.4. Definiciones y acrónimos

A continuación, se definen aquellos términos, tecnicismos, siglas y abreviaciones que figuran en el documento que, por su naturaleza, es difícil explicar por sí mismos. Cuando dichos conceptos aparezcan serán mostrados en cursiva (p.e.: *Unix*), de esta forma se indicará que la definición de dicho termino se especifica en este apartado:

- **Apache license 2.0:** licencia de código *open source* que permite usar, copiar, modificar y distribuir el código que este bajo la misma.
- **Business intelligence*:*** aplicación que usa datos masivos y la estadística. Por este medio permite descubrir información que es difícil de detectar utilizando correlación de datos y con ello se obtiene información adicional que a simple vista es imposible de detectar, reconociendo ciertos patrones que permiten generar modelos estadísticos.
- **Clave-valor:** se refiere al uso de un par de elementos donde el primero indica el contexto del segundo y el segundo indica el valor numérico o semántico del primero.
- **Click:** acción de pinchar el botón de selección del ratón, generalmente el izquierdo para las personas derechas y el derecho para las personas zurdas.
- **Terminal:** es un programa que permite interactuar con un sistema operativo sin necesidad de usar una interfaz gráfica.
- **Cluster:** conjunto de ordenadores conectados entre sí que interaccionan como si fuesen un solo ordenador, se usa para buscar soluciones a problemas *horizontalmente escalables.*
- **Cloud**: se refiere a un sistema de almacenamiento o procesamiento por medio de un *cluster* online.
- **Daemon (demonio):** es un componente software que se instala en un sistema operativo para que cumpla una función específica fuera del entorno gráfico del mismo. Suele ofrecer o consumir servicios al sistema operativo o a la red en la cual está conectado.
- **Dashboard (tablero):** se refiere a un espacio de trabajo en el cual se dispersa un conjunto de tráficos y datos que permiten monitorizar grupos y flujos de información.
- **Debian:** Sistema operativo de código abierto, *Linux*, basado en *UNIX*. Ha sido desarrollado y mantenido por la comunidad de usuarios y desarrolladores de *GNU*.
- **DDoS:** Ataque de denegación de servicios distribuido, es un tipo de ataque informático que es capaz de bloquear un *servidor* por saturación al generar múltiples peticiones de servicio concurrentes.
- **DHCP:** protocolo de comunicación que entrega direcciones IP disponibles en un rango especifico bajo demanda.
- **Diagrama de casos de uso:** diagrama explicativo que ejemplifica la interacción entre el sistema y un actor. La interacción tiene consecuencias que están detalladas en la tabla del caso de uso anexa a cada uno.

- **Escalabilidad horizontal, horizontalmente escalable:** un problema *horizontalmente escalable* es aquel que por su naturaleza no puede ser tratado ampliando la potencia de computo de la máquina que hace el trabajo. Para poder tratar este tipo de problemas es necesario utilizar sistemas distribuidos que permitan distribuir la carga de trabajo entre los distintos *nodos* del sistema, de esta forma, una gran carga de trabajo es repartida entre una cantidad de máquinas que separadamente pueden resolver el problema. Una solución escalable horizontalmente es aquella que permite tratar problemas *horizontalmente escalables.*
- **Escalabilidad vertical, verticalmente escalable:** un problema *verticalmente escalable* es aquel que puede ser solucionado ampliando la capacidad de computo de la máquina que recibe la carga de trabajo. Una solución *verticalmente escalable* es aquella que permite tratar un problema *verticalmente escalable* por medio de la ampliación de los recursos de la máquina.
- **Fichero de log, log, logs:** fichero que contiene registros de una máquina. Dicha información se usa para poder tener un histórico de los procesos y procedimientos que ejecuta una maquina en un fragmento de tiempo.
- **Freeware:** software de prueba de treinta o sesenta días, puede llamarse también versión de prueba o versión trial.
- **GitHub:** es una plataforma de distribución de software que permite subir proyectos para su distribución y desarrollo distribuido de los mismos. También es usado como repositorio.
- **GNU:** proyecto *GNU*, es una iniciativa de software libre iniciada por Richard Stallman. Es propietaria de patentes que permiten la creación, modificación y libre distribución de software de forma gratuita (*GPL*).
- **GPL:** *GNU* General Public Licence (Licencia pública general de *GNU*) es una licencia de uso pública de software que garantiza a los usuarios que dicho sistema software es libre de usar, estudiar, compartir, copiar y modificar.
- **Iptables:** cortafuegos integrado en el *kernel* de *Linux* que permite filtrar los paquetes de datos IP que salen y entran a una maquina en función de las condiciones que se especifiquen.
- **ISO/IEC:** Organización Internacional de Normalización/ Comisión Electrónica Internacional.
- **JSON:** formato de documentos *clave-valor* que contiene información machine readable.
- **Kernel:** núcleo del sistema operativo, elemento de software que inicia las características elementales de un sistema operativo.
- **LAN (red de área local):** Red de área local, red que por sus características abarca una empresa o una organización.
- **Linux:** por lo general hace referencia a un conjunto de sistemas operativos con *kernel* basado en *UNIX*, un ejemplo es *Debian*.
- **Loguear, loguearse, log-in:** acción de iniciar sesión en el sistema por medio de autenticación con usuario y contraseña.
- **Machine readable:** se refiere a usar un formato estandarizado que permita que una maquina sea capaz de interpretar el contenido de un conjunto de datos

por medio del uso de códigos. Estos códigos le "dicen" a la maquina a que pertenece cada dato. Un ejemplo de fichero *machine readable* es aquel que está en formato *XML*

- **MAN (red de área media):** red corporativa de área media, es más grande que una *LAN.*
- **MySQL:** Es un *SGBD* que usa una variante de *SQL* para funcionar.
- **Machine learning:** serie de algoritmos de aprendizaje automático supervisado que permiten obtener modelos estadísticos.
- **Navegador web:** software que por medio de los protocolos HTTP y TCP/IP es capaz de mostrar páginas web.
- **NoSQL:** termino que hace referencia a Not Only SQL (no solo SQL). Es un conjunto de elementos que permiten hacer *persistencia* y recuperación de información por medio de una variación de *SQL*.
- **Nodo, Nodos:** elemento de una red de ordenadores que se encarga de una tarea delegada. Existen 2 tipos: maestro y esclavo. Maestro es el que delega tareas y esclavo el que las cumple.
- **Open source:** dícese del software de código abierto que se puede usar, copiar, modificar y distribuir de forma gratuita. Normalmente va relacionado con las licencias *GPL* o *Apache*. Para efectos de este documento, asociaremos el termino *open source* a sistemas software de licencia gratuita.
- **Ordenadores comodity:** termino que hace referencia a ordenadores de bajo coste y baja potencia de cálculo, estos ordenadores se usan entre sí para crear *clusters*.
- **Parseo, parsear, parseando, parseados:** termino informático anglosajón que nace de la palabra inglesa parse (que traducido es "analizar"). Se refiere a dar formato a un conjunto de datos con tal de que estos puedan ser recuperados por separado.
- **PHP:** lenguaje de programación que permite desarrollar páginas webs con contenido dinámico.
- **Persistencia:** en informática se refiere a almacenar, normalmente en una base de datos. Cuando un sistema es capaz de recuperar información de alguna parte incluso después de haberse apagado, se dice que dicho sistema es persistente.
- **Plugin, Plugins:** componente software que amplía las características base del sistema donde es aplicado.
- **Query:** dícese de una consulta en lenguaje *SQL* o sus derivados.
- **Round Robbin:** técnica que permite rotar una lista de elementos con tal de repartir la carga de trabajo entre ellos o de permitir la sobre escritura del elemento más antiguo para continuar funcionando.
- **Script:** conjunto de instrucciones que se ejecutan en una *terminal* que permiten realizar una acción.
- **Servidor:** elemento de software que se implanta en una maquina con tal de otorgarle las capacidades necesarias para servir información a otras máquinas.
- **Servidor web:** *servidor* que es capaz de enviar o servir páginas web.
- **SGBD (sistema gestor de bases de datos):** Sistema que permite gestionar el uso de bases de datos, una base de datos es u

- **S.O., SS.OO.:** Sistema operativo, o en su plural, sistemas operativos.
- **SQL:** termino que hace referencia a Standart Query Languaje (lenguaje estándar de consultas). Es un lenguaje de programación estandarizado por la norma *ISO/IEC* 9075-1:2008, es un lenguaje de consulta, inserción, modificación y eliminación de datos.
- **Sistema transaccional:** sistema software que por medio de *SQL* permite *persistir* información en una base de datos.
- **SSH:** protocolo de conexión remota cifrado con el protocolo de conexión por capa de puertos seguros (SSL).
- **Unix:** Sistema operativo desarrollado en el M.I.T. en 1960, entre sus particularidades podemos destacar que es multitarea y multiusuario.
- **Windows:** Sistema Operativo de la compañía Microsoft.
- **XML:** formato de archivos *machine readable* que contiene datos y meta datos.
- **WAN (Red de área amplia):** red corporativa o publica de área grande, es más grande que una *MAN.*
- **URL (Uniform Resource Locator):** un localizador de recursos uniforme es una dirección asociada a un recurso web, básicamente es una dirección de una página web.

2. Estado del arte

En este apartado analizaremos las distintas herramientas que ofrece el mercado del software como posibles soluciones. **Haremos una introducción a cada uno de ellos y revisaremos sus beneficios e inconvenientes en una tabla comparativa** que será usada más adelante para decidir cuál de ellas se adapta mejor a las necesidades de la Universidad.

Es importante destacar que para efectos de la metodología de trabajo **métrica V3** este capítulo es formalmente reconocido como Estudio de Viabilidad del Sistema o Estudio de Mercado. Por este motivo, en la planificación de este documento puede apreciarse que el presente capítulo ha sido nombrado como **Estudio de Mercado**.

2.1. Herramientas de monitorización

Luego de realizar el correspondiente estudio de mercado, identificando las distintas herramientas que pueden competir como posible solución al problema, se ha llegado a la conclusión de que las que más se ajustan son las siguientes:

2.1.1 Munin

Munin (2) es un sistema de monitorización de recursos que permite analizar el estado de un grupo de ordenadores conectados a una red LAN. Su trabajo es recuperar ficheros de *log* en los ordenadores monitorizados, obtener información relevante por medio de un sistema de reconocimiento de expresiones regulares, compara esa información con otras y la envía a un *servidor* Munin.

Ilustración 1: Logo de Munin

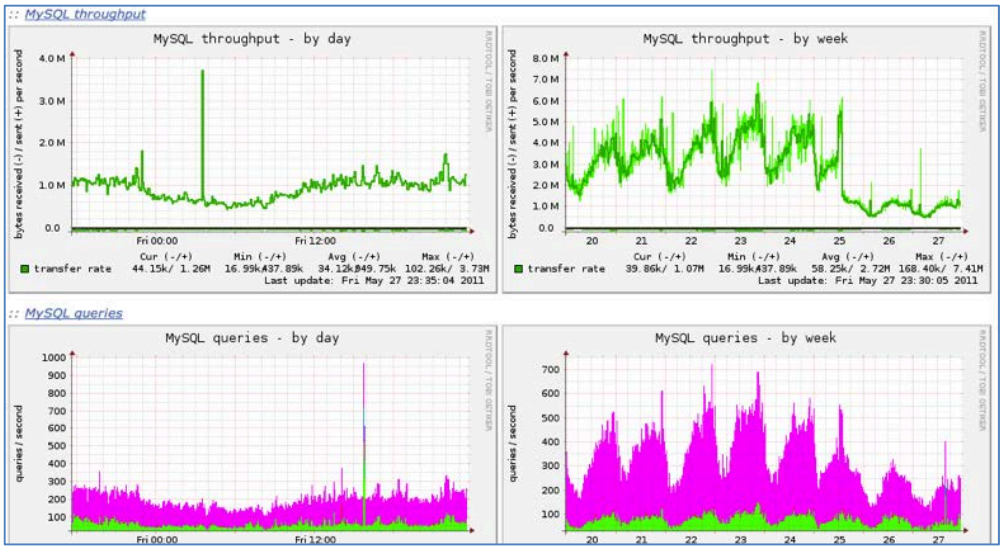


Ilustración 2: Interfaz gráfica de Munin.

El *servidor* Munin recibe la información y dibuja en pares *clave-valor*, con ello puede generar gráficos bidimensionales que permiten tener información histórica de los distintos valores que toma el uso de recursos en función de la clave. Su licencia es *GPL*, por lo cual el uso del software es completamente gratuito. El consumo de recursos (*CPU*, *RAM*) del Munin es razonablemente bajo, lo cual lo convierte en una herramienta viable para monitorizar una red pequeña/mediana.

Munin permite la inclusión de *plugins,* esto permite ampliar la usabilidad del software, todo bajo el concepto de devolver información *clave-valor* desde el ordenador monitorizado hasta el *servidor* que monitoriza.

El problema es que Munin no permite *escalabilidad horizontal* para problemas complejos que deben ser tratados de forma distribuida, por ende, no es viable el uso de este software para monitorizar redes *MAN* o redes *LAN* de gran tamaño.

Actualmente la Universidad cuenta con este sistema, por lo que se incluye en este documento solo para que pueda entrar en la comparativa y ayudar a comprender al lector cuanto mejora la implantación de una nueva solución.

Para más información se recomienda visitar la página oficial de la guía deMunin (3).

2.1.2 Nagios

Nagios (4), o Nagios Core (5) es un sistema de monitorización de recursos muy similar a Munin. Trabaja bajo el mismo concepto, pero también permite supervisión automática y continua de uso de recursos de ordenadores por medio de una red *LAN*.

Ilustración 3: Logo de Nagios.

Su funcionamiento se parece bastante al de Munin, pero tiene funciones de acceso remoto, almacenamiento en base de datos y ciertas funciones distribuidas que Munin no implementa, por lo cual podemos decir que es ligeramente más avanzado.

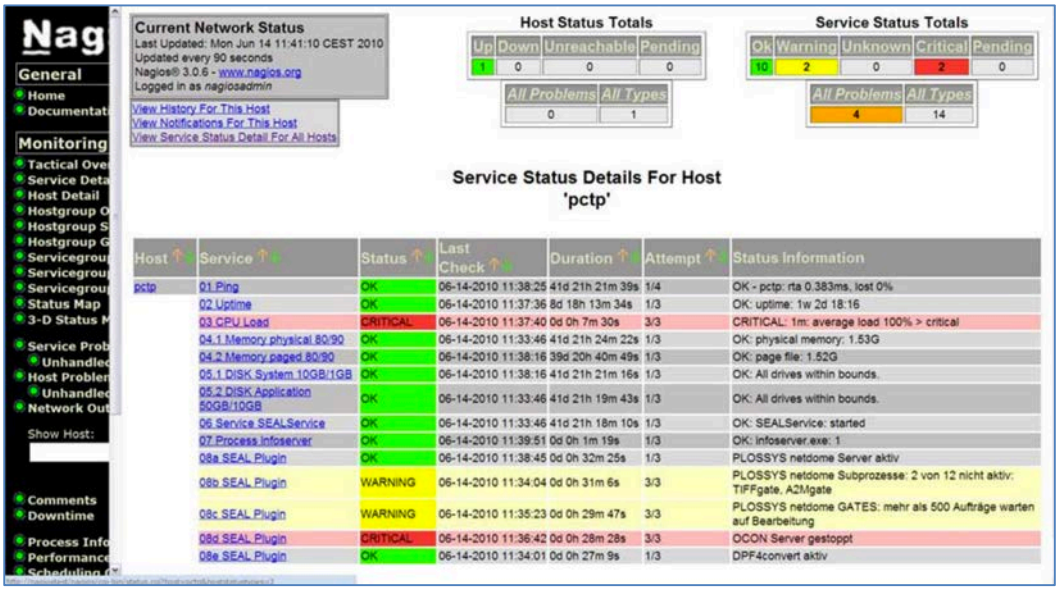


Ilustración 4: Interfaz gráfica de Nagios.

La arquitectura de Nagios se centra, al igual que Munin, en el uso de *plugins*. Estos permiten que sea escalable a la hora de ampliar necesidades de monitoreo por medio de la instalación o creación de los mismos.

Al igual que Munin, la licencia de uso de Nagios es *GPL* y el consumo de recursos es bastante bajo, además, de ser necesario este consumo puede ser distribuido entre distintos *nodos*, de esa forma cada *nodo* recupera la información que le corresponde y la almacena.

El problema en el caso de Nagios se centra en lo mismo: la falta de *escalabilidad horizontal* del sistema, en este caso el de *persistencia*, esto es debido a que el *SGBD MySQL* nos deja persistir y consultar los datos, pero es un *sistema transaccional* clásico y es imposible usarlo de forma distribuida.

Para más información se recomienda visitar la página oficial de Nagios (6), también podemos ver una introducción a Nagios y cómo funciona en el tutorial de nagios-cl.org (7)

Por los requisitos que debe cumplir la solución no revisaremos la versión comercial de Nagios (Nagios XL (8)). Esta versión es de pago y necesitamos claramente una solución *open source* de licencia gratuita.

2.1.3 Ganglia

Ganglia (9) es un sistema de monitorización similar a las soluciones anteriores. Funciona con *nodos* que contienen *demonios* los cuáles recuperan la información de los *logs*, las envían por medio de la red a un *nodo* maestro, este recibe la información y la almacena en una base de datos que para no ser saturada usa un sistema *round robbin*. Por medio de un *servidor web* Apache ofrece la visualización de los gráficos que permiten monitorizar los recursos.

Ilustración 5: Logo de Ganglia.

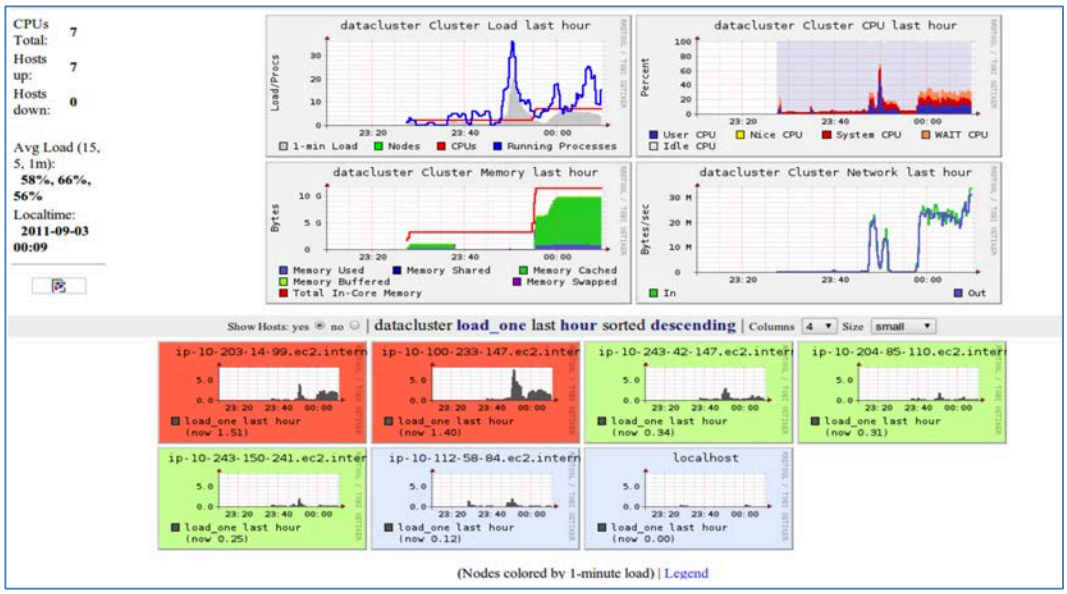


Ilustración 6: Interfaz gráfica de Ganglia.

Ganglia es *multiplataforma*, lo cual facilita mucho su uso en entornos de trabajo con distintos *sistemas operativos*, la base de datos permite *querys* por medio de *scripts PHP* y además permite el uso de *plugins*, los más comunes pueden encontrarse en plataformas como *GitHub* (10).

Para más información se recomienda visitar la página oficial de Ganglia (11)

2.1.4 Splunk

Splunk es un sistema de monitorización avanzado que permite recuperar, *parsear* y almacenar masivamente *logs* de distintas maquinas, enviar dichos *logs* a una base de datos *horizontalmente escalable* y luego mostrar dichos datos por medio de una interfaz gráfica sencilla y amigable.

Ilustración 7: Logo de Splunk.

Splunk funciona muy bien en entornos grandes, permite obtener información útil de los *logs*, sus herramientas más avanzadas facilitan trabajar en la nube de forma automatizada y al mismo tiempo poder analizar dicha información. El problema de este sistema es que la licencia es de pago y la versión *freeware* tiene limitaciones importantes que obligan a optar por otra solución.

Ilustración 8: Interfaz gráfica de Splunk.

Para más información se recomienda visitar la página oficial de Splunk (12).

2.1.5 ELK

ELK (Elasticsearch, Logstash y Kibana) (13) es un sistema compuesto de tres softwares que interaccionan entre sí de forma eficiente. Este sistema permite la recuperación de *logs*, el *parseo* de los mismos, la recuperación y estandarización de las fechas usando expresiones regulares, el almacenamiento distribuido, consultas de lo que hay almacenado y visualización en tiempo real o semi-tiempo real de toda la información almacenada en él.

Ilustración 9: Logo de ELK.

Si bien ELK no es tan potente como Splunk, es una alternativa bastante flexible y completa, lo cual permite adaptarla a las necesidades de distintos entornos de forma eficiente y rápida. Esto lo hace el mejor candidato de este apartado a ser nuestra solución definitiva.

Elasicsearch es un *servidor* de búsqueda e indexado distribuido similar a una base de datos *NoSQL*, que trabaja con registros *JSON*. Este sistema permite una *escalabilidad horizontal* usando *ordenadores comodity*, además permite por medio de *querys* en formato *JSON* hacer consultas robustas y obtener los resultados en un formato que si bien es *machine readable* también es perfectamente legible por humanos.

Logstash es un software de recuperación y *parseo* de *logs* que permite recuperar distintos formatos de fecha y estandarizarlos en un mismo formato único, además, permite limpiar los registros de palabras sin contenido semántico y almacena los datos relevantes del *log* en documentos *JSON* con pares *clave-valor*, al unir ordenadamente los términos es posible recuperar el *log* completo.

Finalmente, Kibana es un software que por medio de *querys* obtiene información de una base de datos o en este caso de Elasticsearch, con la intención de dibujar de distintas formas el avance temporal de los datos obtenido de las máquinas de la red.

ELK es la combinación e integración de estos tres sistemas software en una sola solución sencilla, flexible, escalable, gratuita y *open source*. Esto lo convierte en el mejor candidato a ser nuestra solución.

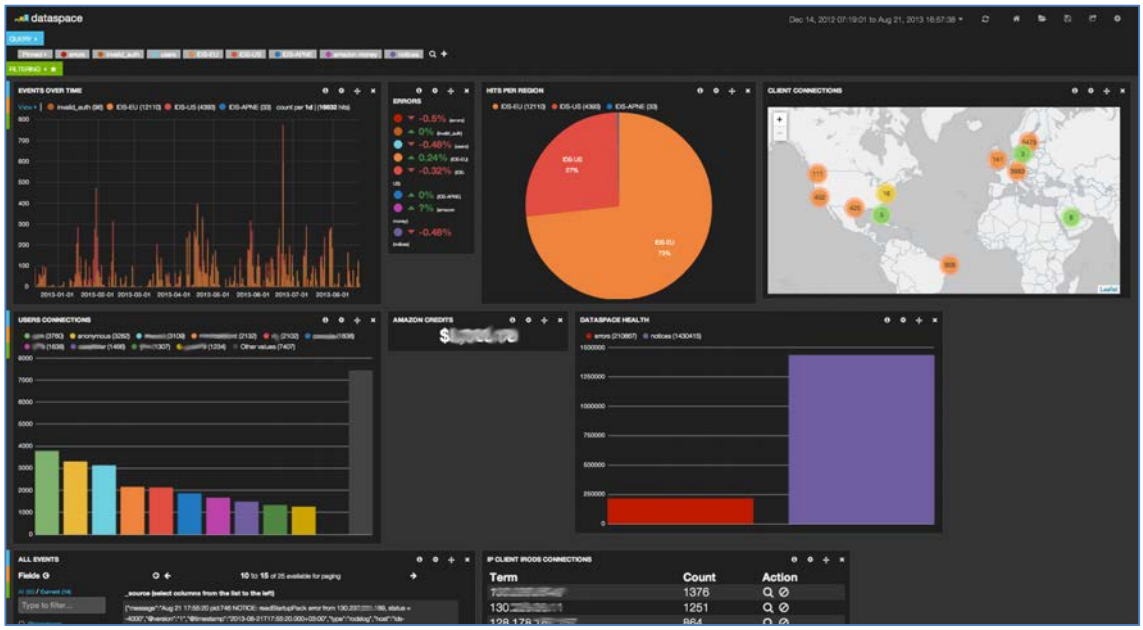


Ilustración 10: Interfaz gráfica de ELK, Kibana.

2.2. Comparativa

En este apartado realizaremos una comparativa profunda sobre los cinco sistemas que hemos visto en el punto 2.1. La finalidad es poder analizar cuál de estos sistemas se adapta más a nuestras necesidades, ya que esto servirá para poder retomar la temática en el capítulo 3 después de haber analizado los requisitos.

Después de un profundo análisis que es aplicable a los cinco candidatos y que al mismo tiempo se enfoca en que elementos son deseables en nuestra solución, la métrica comparativa que usaremos para comparar los cinco sistemas es:

1) ***Escalabilidad horizontal*:** el sistema permite el uso de un *cluster* de ordenadores para solucionar problemas de falta de recursos de forma robusta y a bajo coste.
2) ***Open source* y gratuito:** el sistema es de uso completamente gratuito.
3) **Estandarización de fechas:** el sistema *parsea* de forma automática los distintos formatos de fecha y usa un formato estándar para almacenar estos datos.
4) **Interfaz gráfica amigable:** el sistema cuenta con una interfaz gráfica fácil de usar y adaptable a dispositivos móviles.
5) **Uso de *plugins*:** el sistema permite crear e insertar *plugins* para poder solucionar necesidades futuras del laboratorio.
6) **Consultas a la base de datos:** el sistema permite hacer *querys* a la base de datos o similar.

Para evaluar dicha métrica usaremos dos valores: SI y NO, estos indican:

- **SI**: el sistema cumple la métrica / se puede usar de esa forma.
- **NO**: el sistema no cumple la métrica / no es usable de esa forma.

Herramienta/métrica	1	2	3	4	5	6
Munin	NO	SI	NO	SI	SI	NO
Nagios	NO	SI	NO	SI	SI	NO
Ganglia	NO	SI	NO	SI	SI	NO
Splunk	SI	NO	SI	SI	SI	SI
ELK	SI	SI	SI	SI	NO	SI

Tabla 1: Comparativa con la métrica seleccionada de las cinco soluciones.

Podemos observar en la comparativa que:

- Splunk cumple muy bien las expectativas como solución, pero al no ser *open source*, es preferible evaluar otras opciones.
- ELK no es tan completo como Splunk (p.e.: no cuenta con sistemas *Cloud*), pero para lo que necesita la Universidad, se ajusta perfectamente.

También, es necesario aclarar sobre la tabla que:

- Ganglia si permite hacer *querys* a la base de datos, pero estas funcionan por medio de *scripts*, lo cual lo hace más complicado.
- ELK no cuenta con la posibilidad de utilizar *plugins*, pero si permite recuperar *logs* de un sistema intermediario entre el *S.O.* y los ficheros de *log*.
- Si bien Munin, Nagios y Ganglia son sistemas muy diferentes, para efectos de las necesidades de la Universidad cumplen la misma función.
- Para efectos de la métrica, ELK y Splunk responden igual a las necesidades de la Universidad, pero nosotros sabemos que el hecho de que Splunk no sea *open source* tiene mucho más peso que el que ELK no pueda usar *plugins*. Al fin y al cabo, sigue siendo viable que el sistema operativo mantenga un *daemon* generando *logs* de lo que sea necesario mientras ELK los consume y recupera la información.

3. Análisis

En este apartado revisaremos los requisitos del sistema que debe cumplir aquel que elijamos como solución a las necesidades de la Universidad. Estos requisitos se han obtenido por medio de entrevistas con el cliente, que en nuestro caso es Don Alejandro Calderón Mateos, tutor de este trabajo, para el Laboratorio del Departamento de Informática. Luego de revisar profundamente estos requisitos se ha elaborado una lista de los mismos que será vista en el punto 3.2 de este documento.

La metodología elegida como marco regulatorio para este documento está inspirada en el estándar para proyectos de software que define la **ESA** (14) y **Métrica versión 3** (15), la cual se basa a su vez en los estándares internacionales ***ISO/IEC* 12207** (16) y ***ISO/IEC* 15504** (17)**.**

La métrica antes mencionada es orientativa, ya que su uso común es la planificación, el desarrollo y mantenimiento de sistemas software, motivo por el cual se usará como estructura general.

3.1. Características de la solución deseada

Siguiendo con la temática vista en el punto 1.2, nuestra solución debe cumplir una serie de características que han sido revisadas en las entrevistas con el cliente. Éstas engloban las necesidades de la Universidad, las necesidades de los técnicos del laboratorio y las especificaciones actuales del lugar donde se implantará dicha solución.

Las características que la solución debe contemplar son:

- El sistema debe recuperar *logs*, almacenarlos y mostrar estadísticas del avance de los valores recuperados en los *logs*.
- El sistema debe permitir buscar información relativa a los *logs*
- El sistema debe permitir la recuperación de los *logs*
- El sistema se debe poder implantar con los recursos actualmente disponibles en el laboratorio del Departamento de Informática de la Universidad.
- El sistema debe ser *horizontalmente escalable*.
- El sistema debe ser *open source.*
- La interfaz gráfica de la solución debe permitir la obtención de la información importante de los *logs*.
- La interfaz gráfica de la solución debe poder ser consultada externamente por medio de un *navegador web*.
- El acceso a la interfaz gráfica debe estar restringido por usuario y contraseña.
- El mantenimiento de la solución debe ser nulo o poco costoso en horas hombre y recursos.

3.2. Requisitos de usuario

Con la finalidad de formalizar las características mencionadas anteriormente, debemos registrar en este documento cuáles serán los usos que contemplara la aplicación y a que requisitos corresponden. Para ello utilizaremos una plantilla que contendrá toda la información relativa a cada uno de los requisitos y será fácil de comprender a simple vista.

Definiremos requisito de capacidad como el conjunto de requisitos que indican las capacidades y funciones que debe cumplir la solución. Además, definiremos requisitos de restricción como el conjunto de requisitos que restringen ciertas funciones de la solución.

La plantilla a utilizar será la siguiente:

Identificador	
Titulo	
Prioridad	
Necesidad	
Verificabilidad	
Estabilidad	
Descripción	
Req. relacionados	

Tabla 2: Plantilla estándar para especificación de requisitos.

En la cual, cada campo especificara la siguiente información:

- **Identificador:** será un código asociado a cada requisito que lo diferenciará unívocamente. La codificación se dará en la forma **RX-NN**, donde:
  - **R**: indica que es un requisito de usuario
  - **X**: si es C, indica que el requisito es de capacidad; si es R, indica que el requisito es de restricción
  - **NN**: indica el numero correlativo del requisito, toma valores entre 01 y 99.

- **Título**: indica una breve descripción del requisito.

- **Prioridad**: especifica el nivel de prioridad del requisito, puede tomar valores:
  - **Alta**: indica que el requisito es de implementación obligatoria, es esencial para el funcionamiento de la plataforma

  - **Media**: indica que el requisito se puede implementar, pero si no se implementa no es imprescindible para el funcionamiento de la plataforma.
  - **Baja**: indica que el requisito es de implementación opcional por parte de la solución.

- **Necesidad**: indica la importancia que tiene el requisito para el cliente.
  - **Alta**: el cliente ha indicado que el requisito debe ser satisfecho.
  - **Media**: el cliente ha indicado que el requisito debe ser satisfecho como segunda prioridad, luego de satisfacer los de prioridad alta.
  - **Baja**: el cliente ha indicado que el requisito es opcional.

- **Verificabilidad**: indica el grado en el que puede constatarse que el requisito está incluido en la solución.
  - **Alta**: el requisito es claramente observable en la solución seleccionada.
  - **Media**: el requisito se puede intuir, pero no es claramente observable por el usuario.
  - **Baja**: el requisito no es observable en la solución, por el usuario.

- **Estabilidad**: indica el grado en el cual el requisito puede sufrir cambios durante el desarrollo del proyecto.
  - **Alta**: el requisito es altamente afectable por los cambios que puede sufrir el proyecto.
  - **Media**: el requisito puede varia de forma directamente proporcional a los cambios en el proyecto.
  - **Baja**: el requisito es bastante estable, es muy difícil que sufra cambios.

- **Descripción**: explicación detallada de las características del requisito.

- **Requisitos relacionados**: indicara con que otros requisitos mantiene una relación directa, es útil para identificar qué requisitos se trazan con qué casos de uso.

A continuación, enumeraremos los requisitos de capacidad que debe implementar la solución para satisfacer a las necesidades del cliente:

RC-01	
Titulo	El sistema leerá los *logs* del uso de recursos.
Prioridad	Alta
Necesidad	Alta
Verificabilidad	Media
Estabilidad	Alta
Descripción	El sistema leerá los *logs* producidos por el *S.O.*, los *daemons* o los *plugins* que residen en los ordenadores que generan información.
Req. relacionados	RC-02, RC-03

Tabla 3: Requisito de capacidad RC-01.

RC-02	
Titulo	El sistema recuperara los términos de los *logs* por separado.
Prioridad	Alta
Necesidad	Media
Verificabilidad	Media
Estabilidad	Alta
Descripción	El sistema recuperara los *logs* y hará el *parseo* de los mismos para obtener los términos del *log* por separado.
Req. relacionados	RC-01, RC-03

Tabla 4: Requisito de capacidad RC-02.

RC-03	
Titulo	El sistema recuperara la fecha de los *logs* y la estandarizara a un solo formato.
Prioridad	Alta
Necesidad	Media
Verificabilidad	Media
Estabilidad	Alta
Descripción	Al hacer el *parse* de los *logs*, el sistema reconocerá la expresión regular de la fecha en los distintos formatos existentes y los presentara en un solo formato que identificaremos como formato estándar.
Req. relacionados	RC-01, RC-02

Tabla 5: Requisito de capacidad RC-03.

RC-04	
Titulo	El sistema almacenara la información en una base de datos
Prioridad	Alta
Necesidad	Alta
Verificabilidad	Alta
Estabilidad	Alta
Descripción	El sistema contará con una base de datos que almacenara los *logs* con su información y los términos de los *logs*. Se accederá a la base de datos por medio de la red *LAN*.
Req. relacionados	RC-08, RR-02, RR-03.

Tabla 6: Requisito de capacidad RC-04.

RC-05	
Titulo	El sistema contara con una interfaz gráfica.
Prioridad	Alta
Necesidad	Alta
Verificabilidad	Alta
Estabilidad	Alta
Descripción	El sistema presentara la información del avance del uso de recursos en forma de gráficos, que pueden ser: de líneas, de barra, de columna o circulares.
Req. relacionados	RC-06, RC-07, RC-08, RC-09, RR-06.

Tabla 7: Requisito de capacidad RC-05.

RC-06	
Titulo	La interfaz gráfica del sistema mostrara el avance del uso de recursos.
Prioridad	Alta
Necesidad	Alta
Verificabilidad	Alta
Estabilidad	Alta
Descripción	El sistema mostrara el avance el uso de recursos en función del tiempo por medio de gráficos de línea.
Req. relacionados	RC-05, RC-07.

Tabla 8: Requisito de capacidad RC-06.

RC-07	
Titulo	El sistema permitirá identificar fallas y caídas de funcionamiento.
Prioridad	Alta
Necesidad	Alta
Verificabilidad	Alta
Estabilidad	Alta
Descripción	Por medio de los gráficos de líneas, el sistema permitirá identificar fallas y caídas en los servicios, mostrando una discontinuidad en el avance de los servicios y permitirá enviar de alertas por correo electrónico.
Req. relacionados	RC-05, RC-06.

Tabla 9: Requisito de capacidad RC-07.

RC-08	
Titulo	La interfaz gráfica del sistema permitirá recuperar datos de la base de datos.
Prioridad	Alta
Necesidad	Alta
Verificabilidad	Media
Estabilidad	Alta
Descripción	La interfaz gráfica del sistema permitirá seleccionar datos y recuperar información de la base de datos por medio del uso de un lenguaje de recuperación de datos, como *SQL* o similar.
Req. relacionados	RC-05.

Tabla 10: Requisito de capacidad RC-08.

RC-09	
Titulo	La interfaz gráfica del sistema será accesible desde fuera de la red.
Prioridad	Alta
Necesidad	Media
Verificabilidad	Media
Estabilidad	Alta
Descripción	La interfaz gráfica del sistema permitirá el acceso remoto desde fuera de la red. El acceso al sistema estará restringido al personal del Laboratorio del Departamento de Informática de la Universidad.
Req. relacionados	RC-05.

Tabla 11: Requisito de capacidad RC-09.

RC-010	
Titulo	El sistema permitirá buscar información durante todo el año
Prioridad	Baja
Necesidad	Media
Verificabilidad	Baja
Estabilidad	Alta
Descripción	El sistema tendrá una disponibilidad de 365 días al año, con un margen de error del 10%.
Req. relacionados	

Tabla 12: Requisito de capacidad RC-10.

RC-011	
Titulo	El sistema permitirá incluir *plugins* para aumentar sus capacidades.
Prioridad	Media
Necesidad	Media
Verificabilidad	Baja
Estabilidad	Alta
Descripción	El sistema permitirá incluir *plugins* que permitan aumentar las capacidades actuales del mismo.
Req. relacionados	

Tabla 13: Requisito de capacidad RC-11.

A continuación, enumeraremos los requisitos de restricción que debe implementar la solución para responder a las necesidades del cliente:

RR-01	
Titulo	Para acceder a la interfaz del sistema es obligatorio iniciar sesión.
Prioridad	Alta
Necesidad	Alta
Verificabilidad	Alta
Estabilidad	Alta
Descripción	El sistema accederá por una dirección IP o una *URL* del Laboratorio y para bloquear el acceso no autorizado pedirá un usuario y una contraseña, las cuáles serán administradas por el personal del Laboratorio del Departamento de Informática de la Universidad.
Req. relacionados	RC-05

Tabla 14: Requisito de capacidad RR-01.

RR-02	
Titulo	La base de datos del sistema será *horizontalmente escalable*.
Prioridad	Alta
Necesidad	Alta
Verificabilidad	Media
Estabilidad	Alta
Descripción	La base de datos debe permitir la ampliación de su capacidad de computo por medio de la instalación de más ordenadores con el mismo sistema en un conjunto *Cluster* trabajando de forma distribuida con el fin se ser *horizontalmente escalable*.
Req. relacionados	RC-04, RR-03.

Tabla 15: Requisito de capacidad RR-02.

RR-03	
Titulo	La base de datos debe ser consultable.
Prioridad	Alta
Necesidad	Alta
Verificabilidad	Media
Estabilidad	Alta
Descripción	La base de datos debe poder permitir extraer información por medio de un lenguaje de consulta y selección de datos, ya sea *SQL* o similar.
Req. relacionados	RC-04, RR-02.

Tabla 16: Requisito de capacidad RR-03.

RR-04	
Titulo	El sistema será *open source*.
Prioridad	Alta
Necesidad	Alta
Verificabilidad	Alta
Estabilidad	Alta
Descripción	El sistema será de código abierto y licencia gratuita, ciñéndose a las licencias de uso *open source* tales como *Apache License 2.0* o *GPL*.
Req. relacionados	

Tabla 17: Requisito de capacidad RR-04.

RR-05	
Titulo	El sistema debe funcionar de forma distribuida con los recursos actuales de la Universidad.
Prioridad	Alta
Necesidad	Alta
Verificabilidad	Alta
Estabilidad	Alta
Descripción	El sistema deberá ser implantarle en el laboratorio del departamento de informática de la Universidad, basándose en los recursos actuales del mismo, tales como, aulas informáticas, ordenadores de las aulas red *LAN*, *servidores* de la Universidad, etc.
Req. relacionados	

Tabla 18: Requisito de capacidad RR-05.

RR-06	
Titulo	El sistema debe enviar la información de forma cifrada por medio del protocolo *SSL*.
Prioridad	Alta
Necesidad	Alta
Verificabilidad	Alta
Estabilidad	Alta
Descripción	El sistema deberá comunicarse de forma segura por medio del *Navegador web* usando el protocolo de capa de sockets seguros *SSL* o *TLS*.
Req. relacionados	RC-05.

Tabla 19: Requisito de capacidad RR-06.

3.3. Determinación del entorno operacional

En este apartado definiremos el entorno operacional en el cual deberá funcionar la solución. En términos generales, el entorno operacional será el laboratorio de informática, el cual usa una serie de recursos que mencionaremos a continuación:

Aulas informáticas:

Las aulas informáticas del Laboratorio del Departamento de Informática son uno de los elementos del entorno operacional, en el cual deberá funcionar el elemento distribuido del sistema que recuperará la información.

Las especificaciones técnicas de los ordenadores de las aulas informáticas son:

Especificaciones técnicas	
Sistema Operativo	*Windows* 7 x64 / *Debian* 8 x64
Procesador	Intel Core I5-4770k 3.5Ghz
Memoria RAM	4GB DDR3 1600Mhz
Disco duro	320GB Sata III
Tarjeta de red	Broadcom BCM5703X Gigabit Ethernet

Tabla 20: Especificaciones técnicas de los ordenadores de las aulas informáticas.

De estos ordenadores se recuperará la información, **por el momento** la implantación del sub-sistema de recuperación de *logs* se hará **únicamente en el sistema operativo *Debian* 8**. La instalación del sub-sistema de recuperación de *logs*, o *daemon*, se hará en las dos aulas informáticas (4.0.F16 y 4.0.F18) las cuáles cuentan con 20 ordenadores cada una, lo cual suma **un total de 40 ordenadores**.

Red local LAN:

La red de área local *LAN* de la Universidad funciona con direcciones IP del rango 163.117.142.XXX, donde XXX corresponde a las direcciones IP disponibles por *DHCP*, las cuáles están entre la 176 y la 199. Estas direcciones IP corresponden a las que están disponibles para asignar al *nodo* maestro de la base de datos de la solución, de esas se nos ha otorgado la dirección **IP 163.117.142.181**. Las direcciones del *Cluster*, de ser necesario, serán internas y serán del tipo Clase C.

Por otro lado, las direcciones IP de las aulas informáticas están asignadas estáticamente por el Laboratorio del Departamento de Informática. Estas direcciones corresponden al rango 163.117.142.XXX, donde XXX toma valores entre 101 y 120 para el aula 4.0.F16 y valores entre 201 y 220 para el aula 4.0.F18.

Los *servidores* DNS y el cortafuego de la Universidad garantizan la respuesta a caídas de servicio y ataques externos en términos de seguridad y estabilidad general. Además, toda la red *LAN* de la Universidad es gigabit ethernet, por lo que el ancho de banda no supone un problema para la poca carga de trabajo que tiene actualmente. Posibles mejoras para la red *LAN* en caso de aumentar el tamaño de ordenadores a monitorizar serán explicadas en el capítulo 8 - “Conclusiones y trabajos futuros”.

***Servidor* de acceso a la interfaz gráfica y *nodo* maestro de la base de datos:**

El *servidor* que nos permitirá acceder a la interfaz gráfica y poder almacenar los datos será el *nodo* maestro del sistema de almacenamientos de datos distribuido de nuestra solución. Dicho *servidor* tomara la IP asignada por el Laboratorio de Informática, **163.117.142.181** y tendrá acceso por: el puerto de comunicaciones HTTPS

443; el puerto de comunicaciones HTTP 80, por el cual solo permitirá iniciar comunicaciones, pero luego deberá redirigirlas para usar SSL; el puerto de conexión segura *SSH* 23 y el puerto que use la aplicación para escuchar las llamadas a la base de datos.

Las especificaciones técnicas del *servidor* son las mismas que las de los ordenadores de las aulas informáticas, éstas se indican a continuación:

Especificaciones técnicas	
Sistema Operativo	*Windows* 7 x64 / *Debian* 8 x64
Procesador	Intel Core I5-4770k 3.5Ghz
Memoria RAM	4GB DDR3 1600Mhz
Disco duro	320GB Sata III
Tarjeta de red	Broadcom BCM5703X Gigabit Ethernet

Tabla 21: Especificaciones técnicas del *servidor* de la solución.

Regulación del entorno operacional:

La comunicación del entorno operacional vendrá dada por la naturaleza de la arquitectura tanto de la red *LAN* como de la solución. La descripción de dicha arquitectura se muestra en el siguiente diagrama:

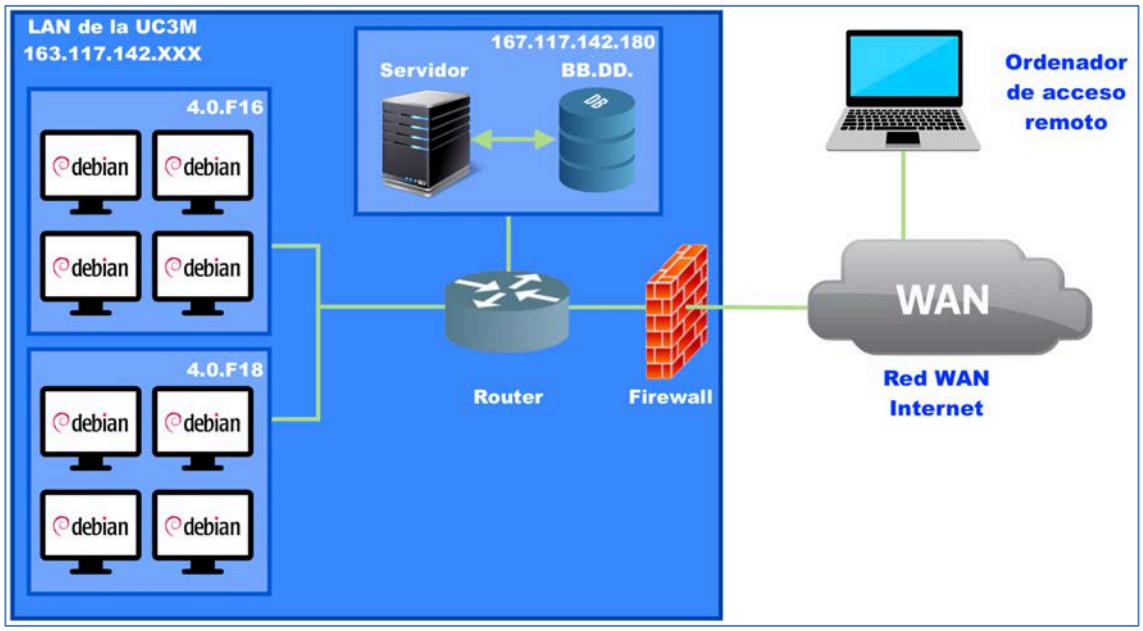


Ilustración 11: Diagrama del entorno operacional de la solución.

En él podemos ver que la red *LAN* de la Universidad trabaja con direcciones IP del rango 163.117.142.XXX, que son direcciones IP públicas y por lo tanto son accesibles desde la red *WAN* o internet, ese es el motivo por el cual es necesario implantar una solución que permita usar *SSL* y acceso con autenticación.

También podemos ver que todas las comunicaciones externas pasan por el cortafuegos de la Universidad, el cual nos protege de ataques externos como *DDoS*, entre otros.

Por último, podemos ver que las aulas de informáticas pueden acceder a la base de datos y al trabajar ambas en la misma red la comunicación es rápida y fluida.

3.4. Trazabilidad de requisitos con las soluciones disponibles

En este apartado analizaremos en profundidad las soluciones y cómo reaccionan frente a los requisitos especificados en el punto 3.2. La tabla de trazabilidad que se muestra a continuación formaliza lo visto en el punto 2.2:

	Munin	Nagios	Ganglia	Splunk	ELK
RC-01	si	si	si	si	si
RC-02	no	no	no	si	si
RC-03	no	no	no	si	si
RC-04	no	no	si	si	si
RC-05	si	si	si	si	si
RC-06	si	si	si	si	si
RC-07	si	si	si	si	si
RC-08	no	no	no	si	si
RC-09	si	si	si	si	si
RC-10	si	si	si	si	si
RC-11	si	si	si	si	No*
RR-01	si	si	si	si	si
RR-02	no	no	no	si	si
RR-03	no	no	no	si	si
RR-04	si	si	si	no	si
RR-05	si	si	si	si	si
RR-06	si	si	si	si	si

Tabla 22: Traza entre requisitos y soluciones disponibles.

Desde esta perspectiva podemos observar que Splunk y ELK son las soluciones que más se ajustan a las necesidades del cliente. Otro aspecto importante a destacar es que la única solución que no es *open source* es Splunk y que ELK no soporta *plugins*.

Se incluye NO* en la tabla resumen para destacar que con ELK es posible incluir filtros (18), los cuáles hacen una función similar a los *plugins*, la diferencia es que estos no funcionan por *bash*, lo cual es un limitante que implica que los técnicos de laboratorio deben adecuarse a este método de trabajo y a las limitaciones implícitas.

3.5. Elección de la solución

Descartando automáticamente las soluciones que nos son *horizontalmente escalables* solo podemos elegir entre dos opciones:

- **ELK:** *horizontalmente escalable*, *open source*, no soporta *plugins*.
- **Splunk:** *horizontalmente escalable*, de pago, soporta *plugins*.

Intentando ajustar la solución ELK como la opción más adecuada, hemos conversado con el cliente y él ha especificado que el peso de no soportar *plugins* no lo descarta del todo, ya que escribir un *script*, o usar un software que genere *logs* no es difícil de solucionar. Al hacer esto, los *logs* serían consumidos por ELK y la información seria recuperable de igual forma.

Por el contrario, el inconveniente de tener que asumir un coste continuo con la compañía Splunk Inc. por el concepto de mantenimiento, implantación y sistemas *cloud* sería inviable para la asignación de recursos que actualmente otorga la Universidad al laboratorio.

Fuera de la comparativa y destacando algunos aspectos de ELK, podemos ver que:

- Es fácil crear un *Cluster*, Elasticsearch busca otros *nodo*s para hacer balanceo de carga automáticamente y crear un índice distribuido.
- Logstash es fácil de instalar, automatizable, y está integrado en la solución.
- Kibana está hecho a medida para Elasticsearch, se conoce como la interfaz gráfica de Elasticseach por defecto.
- ELK es tanto *horizontalmente escalable* como *verticalmente escalable*.
- Existen herramientas que permiten conectar con ELK para ampliar sus capacidades, tales como Beats Platform o Prelert.
- Kibana funciona por medio del *servidor* web Apache, lo que nos permite añadir buen rendimiento y seguridad a la interfaz gráfica.

Por todas estas razones, **la solución elegida es la suite ELK**. De ahora en adelante nos referiremos a ELK como “el sistema” integrando automáticamente en el término a sus 3 componentes: Logstash, Elasticsearch y Kibana.

4. Diseño

En este apartado revisaremos cual va a ser la arquitectura que se ha diseñado para el sistema, cuáles serán los flujos de información, como se visualizará la interfaz gráfica y cómo funcionará el sistema en el entorno operacional.

4.1. Definición de la arquitectura del sistema

La idea principal de este apartado es poder presentar cual será la arquitectura del sistema y cuál será el flujo de información desde que se recoge del *log* hasta que se presenta en el navegador. En la siguiente ilustración podemos ver un diagrama que explica el flujo de información y las distintas etapas que esta cruza.

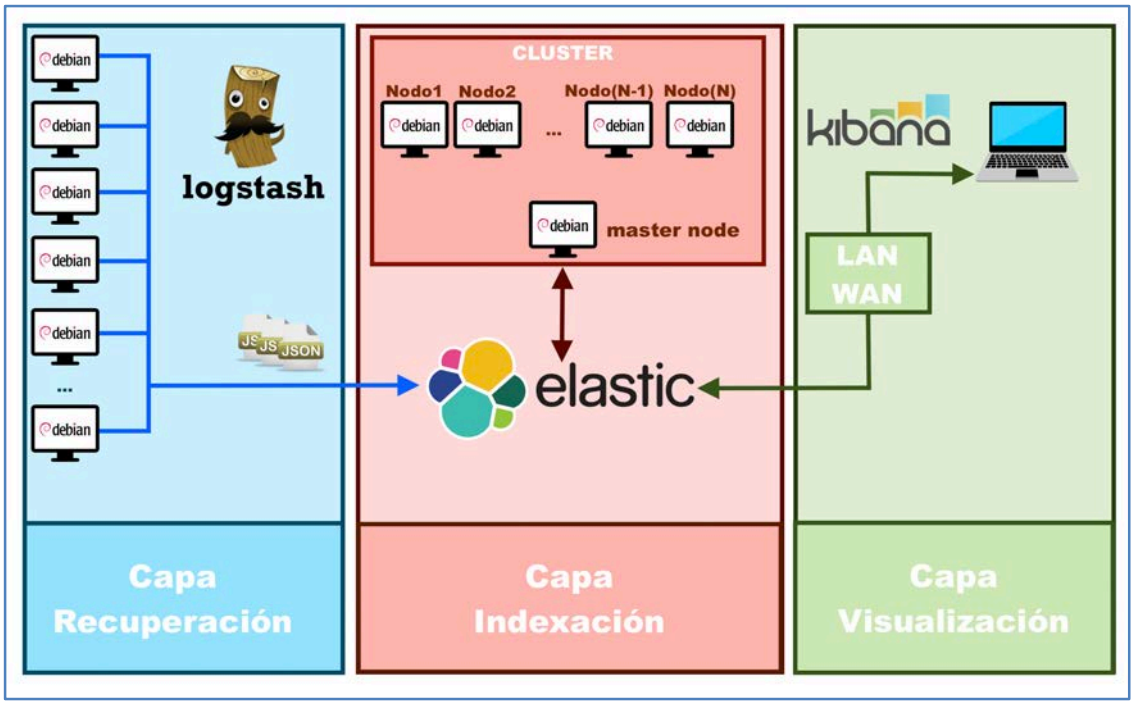


Ilustración 12: Arquitectura de ELK y nombres de las capas.

Como se puede ver en la ilustración anterior, este proyecto utilizara tres capas distintas, propias de la arquitectura de ELK:

- **Capa de Recuperación:** esta capa se encarga de recuperar la información directamente de la fuente, *parsear* esta información y entregarla en formato *JSON* a la siguiente capa.

- **Capa de Indexación:** esta capa se encarga de recuperar, almacenar e indexar los ficheros recuperados de la primera capa. La idea principal es poder contar con un *clúster* de N *nodos* que responda rápidamente a las *querys* y que indexe por medio de un *nodo* maestro.

- **Capa de Visualización:** esta última capa se encarga de recuperar la información de la primera capa por medio de *querys* y presentarla en forma de gráficos. En esta capa podemos hacer nuestro *bussines intelligence*.

Para efectos de este documento, el *clúster* no existe como tal, trabajaremos solo con el *nodo* maestro, ya que es una idea que se sale de los márgenes de tiempo establecidos, sin embargo, se comenta como una opción de escalabilidad viable a futuro dentro del **Capítulo 8, punto 2 – Trabajos futuros.**

4.2. Diseño de casos de uso

En este apartado definiremos en qué casos el usuario o el cliente interaccionará con la aplicación y cuáles serán los resultados de esa interacción, según se define en **métrica versión 3**, la nomenclatura es **casos de uso**. Para ello definiremos una tabla que, como en el apartado anterior, contendrá la información del caso de uso y será fácil de comprender a simple vista.

La plantilla que usaremos será la siguiente:

Identificador	
Titulo:	
Actores:	
Objetivo:	
Escenario:	
Precondiciones:	
Postcondiciones:	

Tabla 23: Plantilla de casos de uso.

En la cual los apartados identifican la siguiente información:

- **Identificador:** permite diferenciar unívocamente al caso de uso mediante un código con la forma **CU-NN**, el cual se compone de:
  - **CU:** indica que el identificador corresponde a un caso de uso.
  - **NN:** indica el numero unívoco del caso de uso.
- **Título:** indica el nombre del caso de uso.
- **Actores:** identifica el rol que juega cada usuario durante la interacción del caso de uso. Puede tomar tres valores distintos.
  - **Usuario:** indica que el actor es el usuario de las aulas informáticas, es decir, generalmente estudiantes que hacen prácticas de asignaturas.
  - **Administrador:** indica que el actor es un técnico de laboratorio, con acceso privilegiado que le permite ver la interfaz gráfica del sistema.

- **Maquina:** indica que el actor es el ordenador en sí mismo, que, por medio del sistema operativo, consume recursos o registra sucesos y esto genera *logs*.

- **Objetivo:** propósito de la interacción del caso de uso.
- **Escenario:** indica los pasos que el actor realiza para completar el objetivo.
- **Precondiciones:** identifica los requisitos que deben cumplirse para que se pueda generar la interacción.
- **Postcondiciones:** indica el estado en el que queda el sistema luego de la interacción.

Para acelerar la comprensión del caso de uso se incluirá un *diagrama de casos de uso* antes de cada caso de uso, con ello pretendemos que el lector pueda comprender rápidamente la interacción entre el usuario y el sistema, identificar a los actores y cuál es el resultado de esa interacción. El termino *diagrama de casos de uso* junto con la nomenclatura se explican en el punto **1.4 - Definiciones y acrónimos** de este documento.

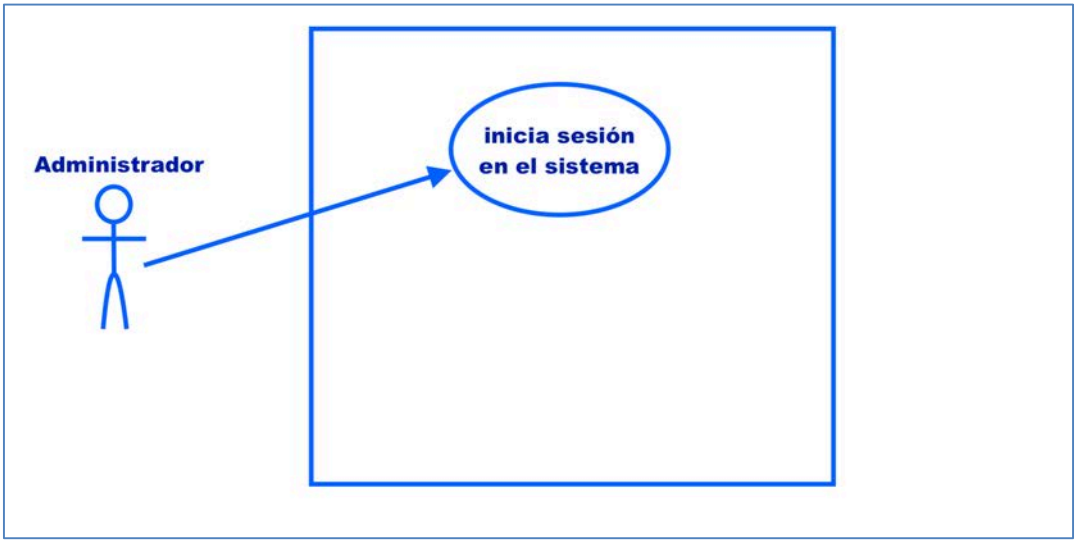


Ilustración 13: Diagrama de caso de uso de CU-01.

CU-01	
Titulo:	Inicio de sesión en el sistema.
Actores:	Administrador.
Objetivo:	Bloquear el acceso no autorizado al sistema y mantener una sesión durante todo el proceso de uso.
Escenario:	El administrador ingresa la *URL* de acceso a la aplicación en el navegador, ingresa su nombre de usuario y contraseña y hace *click* en el botón “iniciar sesión”.
Precondiciones:	El sistema tiene que estar operativo, la red *LAN* de la Universidad tiene que estar operativa.
Postcondiciones:	El usuario puede hacer uso de los servicios del sistema mientras se mantenga la sesión iniciada.

Tabla 24: Caso de uso CU-01.

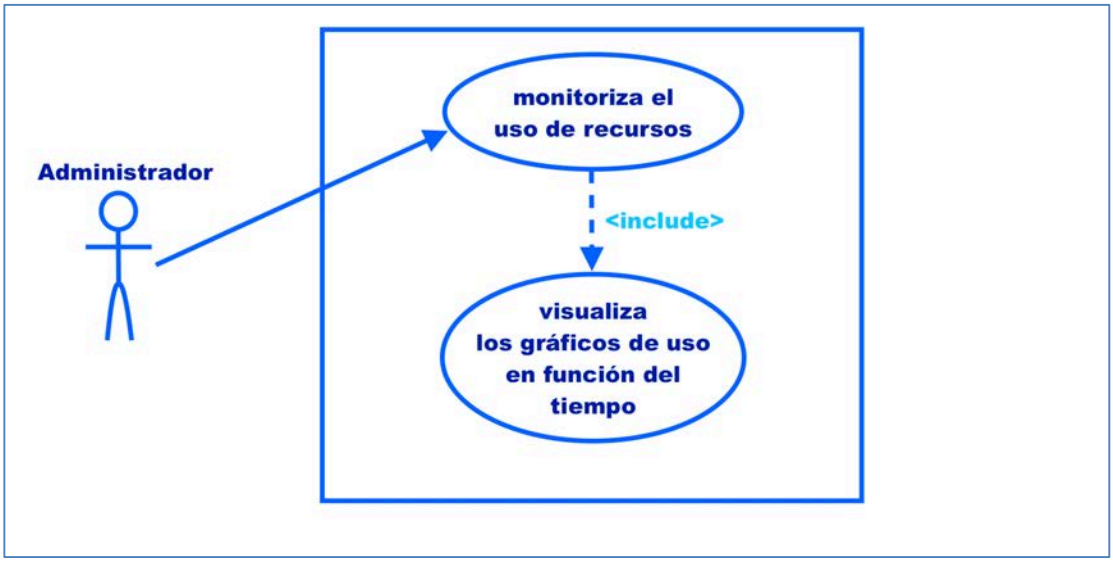


Ilustración 14: Diagrama de caso de uso de CU-02.

CU-02	
Titulo:	Monitorización de uso de recursos.
Actores:	Administrador.
Objetivo:	Observar el comportamiento del uso de recursos en función del tiempo.
Escenario:	Luego de iniciar sesión, el usuario debe buscar y seleccionar la opción de monitorización de recursos.
Precondiciones:	El usuario debe haber iniciado sesión, el sistema debe contener datos del uso de recursos.
Postcondiciones:	El sistema muestra la información del uso de recursos en función del tiempo en gráficos bidimensionales. Los gráficos se actualizan a medida que pasa el tiempo.

Tabla 25: Caso de uso CU-02.

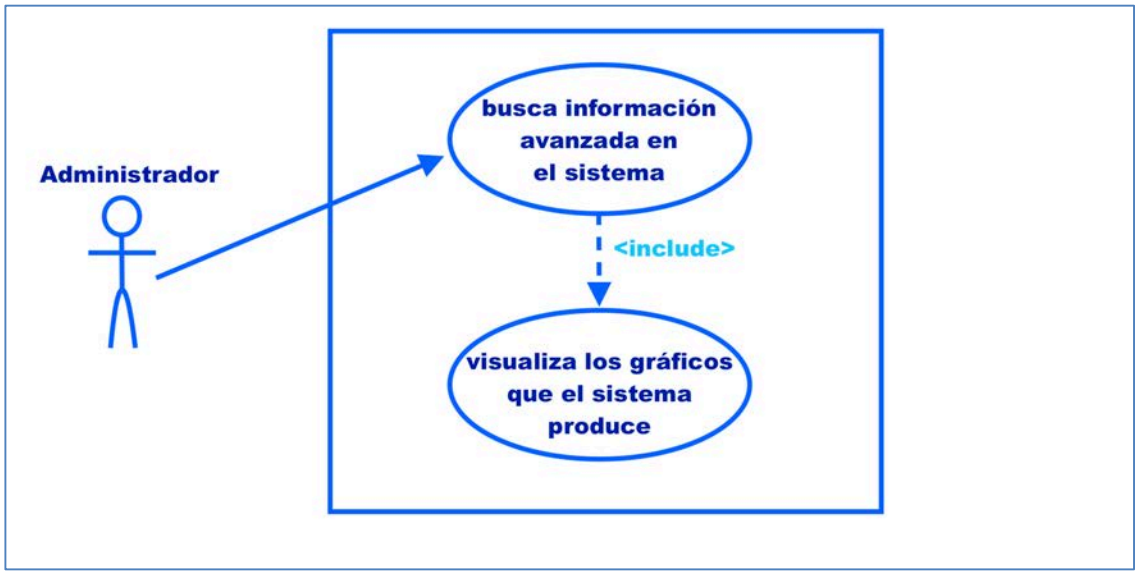


Ilustración 15: Diagrama de caso de uso de CU-03.

CU-03	
Titulo:	Búsqueda avanzada en el sistema.
Actores:	Administrador.
Objetivo:	Buscar información relativa al uso de recursos de forma avanzada lanzando consultas al sistema.
Escenario:	Luego de iniciar sesión, el usuario debe buscar y seleccionar la opción de búsqueda avanzada, insertar la consulta y visualizar los resultados.
Precondiciones:	El usuario debe haber iniciado sesión, el sistema debe contener datos del uso de recursos.
Postcondiciones:	El sistema recupera la información consultada y muestra la información del uso de recursos en función del tiempo en gráficos bidimensionales. Los gráficos se actualizan a medida que pasa el tiempo.

Tabla 26: Caso de uso CU-03.

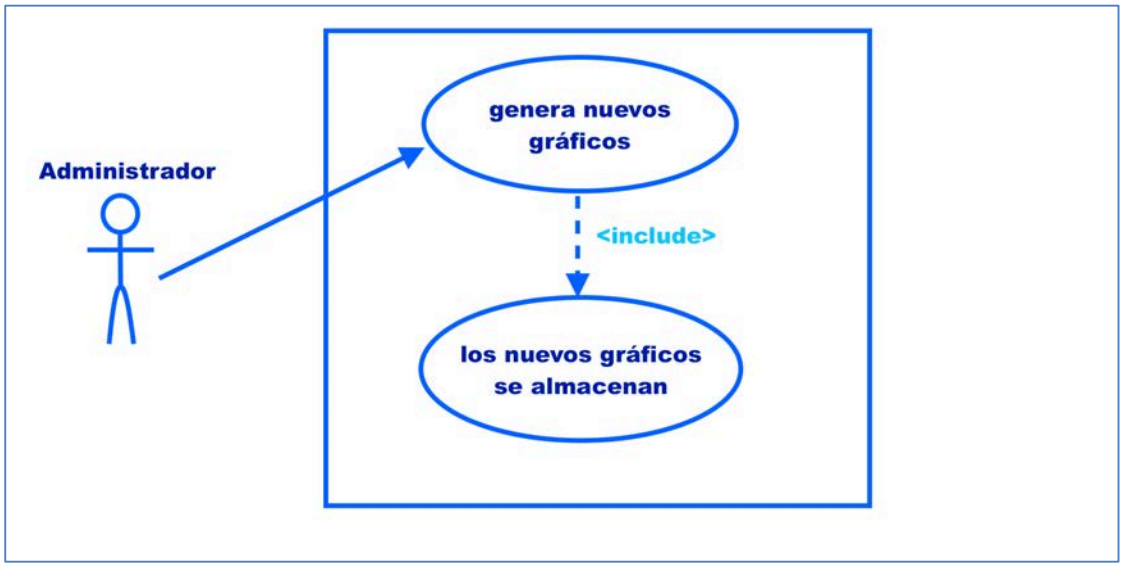


Ilustración 16: Diagrama de caso de uso de CU-04.

CU-04	
Titulo:	Generación de nuevos gráficos de monitorización.
Actores:	Administrador.
Objetivo:	Luego de hacer una búsqueda avanzada, se almacenara esa instancia para ser consultada más adelante.
Escenario:	El usuario debe crear una vista de la búsqueda avanzada que se ha hecho previamente y guardarla en el sistema.
Precondiciones:	El sistema debe contener datos de una búsqueda avanzada reciente.
Postcondiciones:	El sistema contiene una nueva vista que permite monitorizar recursos por medio de gráficos bidimensionales, el usuario puede visualizarlos cuando desee.

Tabla 27: Caso de uso CU-04.

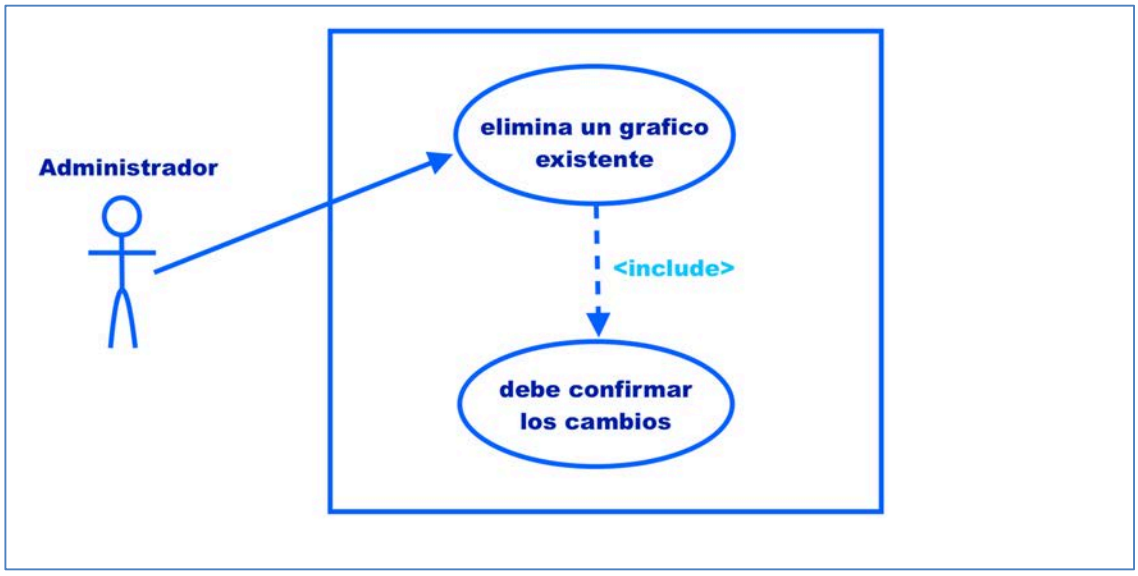


Ilustración 17: Diagrama de caso de uso de CU-05.

CU-05	
Titulo:	Eliminación de un gráfico existente.
Actores:	Administrador.
Objetivo:	Luego de existir una instancia de una consulta existente, esta será eliminada del sistema.
Escenario:	El usuario debe seleccionar la vista que desea borrar y luego eliminarla, el sistema pedirá confirmación de la eliminación.
Precondiciones:	El sistema debe contener datos de al menos un gráfico de monitorización.
Postcondiciones:	El sistema contiene una vista de gráficos de monitorización menos que antes.

Tabla 28: Caso de uso CU-05.

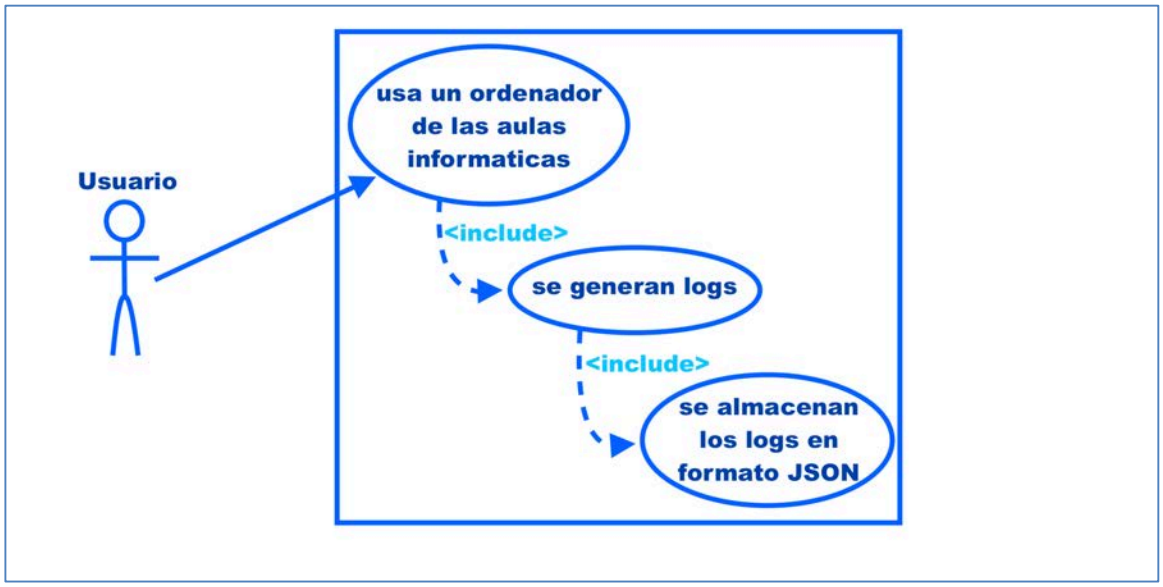


Ilustración 18: Diagrama de caso de uso de CU-06.

CU-06	
Titulo:	Generación y consumo de *logs* por uso del usuario.
Actores:	Administrador.
Objetivo:	Almacenar la información relativa a que recursos son usados por el usuario y en qué momento.
Escenario:	A medida que el usuario use los recursos de la máquina, ésta generará *logs* los cuáles serán recuperados por el sistema, *parseados* y enviados en formato *JSON* al *servidor* Elasticsearch.
Precondiciones:	El ordenador que será usado por el usuario debe contener el elemento Logstash del sistema para poder recuperar la información.
Postcondiciones:	El sistema contiene nueva información sobre el uso de recursos, esta información esta almacenada en el sistema de forma persistente.

Tabla 29: Caso de uso CU-06.

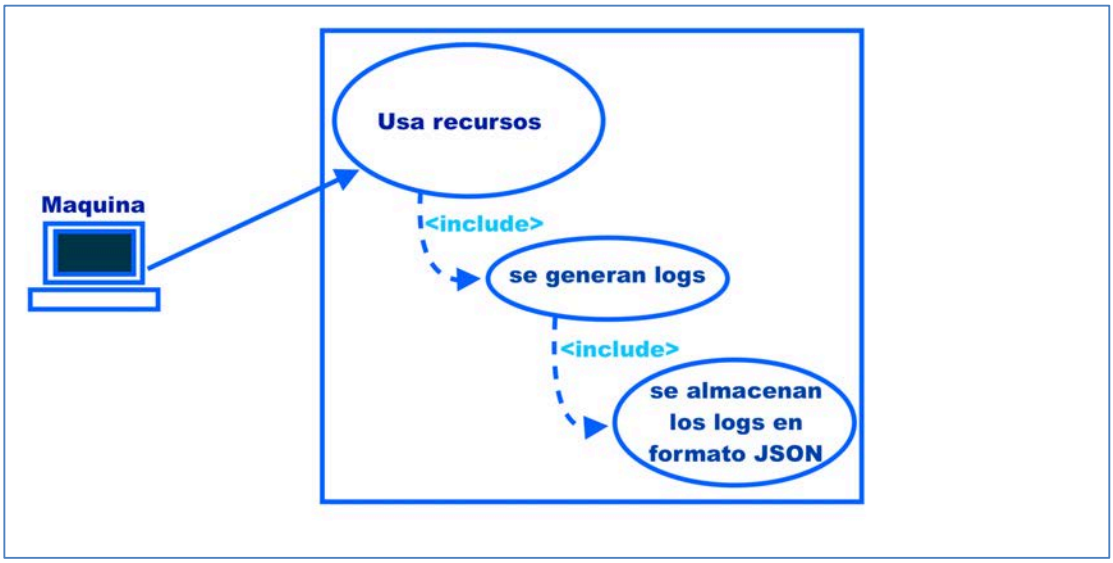


Ilustración 19: Diagrama de caso de uso de CU-07.

CU-07	
Titulo:	Generación y consumo de *logs* por uso de recursos por la máquina.
Actores:	Maquina.
Objetivo:	Almacenar la información relativa a que recursos son usados por la máquina y en qué momento.
Escenario:	A medida que la maquina use los recursos sus propios recursos, ésta generará *logs* los cuáles serán recuperados por el sistema, *parseados* y enviados en formato *JSON* al *servidor* Elasticsearch.
Precondiciones:	El ordenador que será usado por el usuario debe contener el elemento Logstash del sistema para poder recuperar la información.
Postcondiciones:	El sistema contiene nueva información sobre el uso de recursos, esta información esta almacenada en el sistema de forma persistente.

Tabla 30: Caso de uso CU-07.

4.3. revisión de la interfaz de usuario

En este apartado revisaremos la interfaz gráfica de Kibana con la intención de poder explicar las distintas funciones que incorpora y como poder acceder a dichas funciones. Cada imagen explicativa contendrá rectángulos en rojo con un número, éstas figuras indicaran los distintos elementos que contiene la interfaz gráfica del sistema.

Página inicial:

Para acceder a la página inicial debemos introducir la *URL* de la aplicación un *navegador web* como Google Chrome. La *URL* que debemos introducir es **http://163.117.142.181/** y luego debemos presionar la tecla INTRO, al hacer esto podremos ver la siguiente ventana:

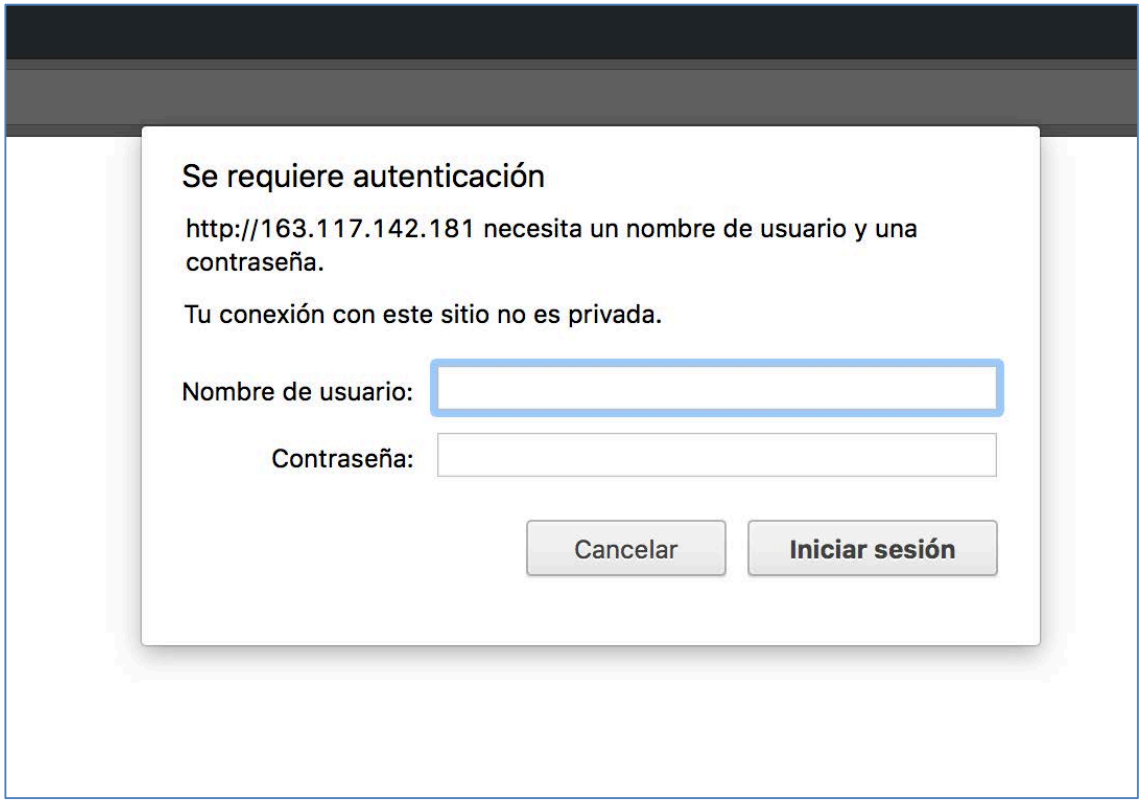


Ilustración 20: Inicio de sesión en el sistema.

Esta ventana restringe el acceso no permitido, para poder seguir avanzando por el resto de las vistas del sistema debemos introducir el usuario y la contraseña, luego, debemos hacer *click* en el botón "Iniciar sesión":

- **Usuario:** "kibana"
- **Contraseña:** "articuno"

La página nos re direccionara automáticamente a la *web* del sistema, donde podremos ver la interfaz de Kibana con sus distintos elementos. La primera vista será:

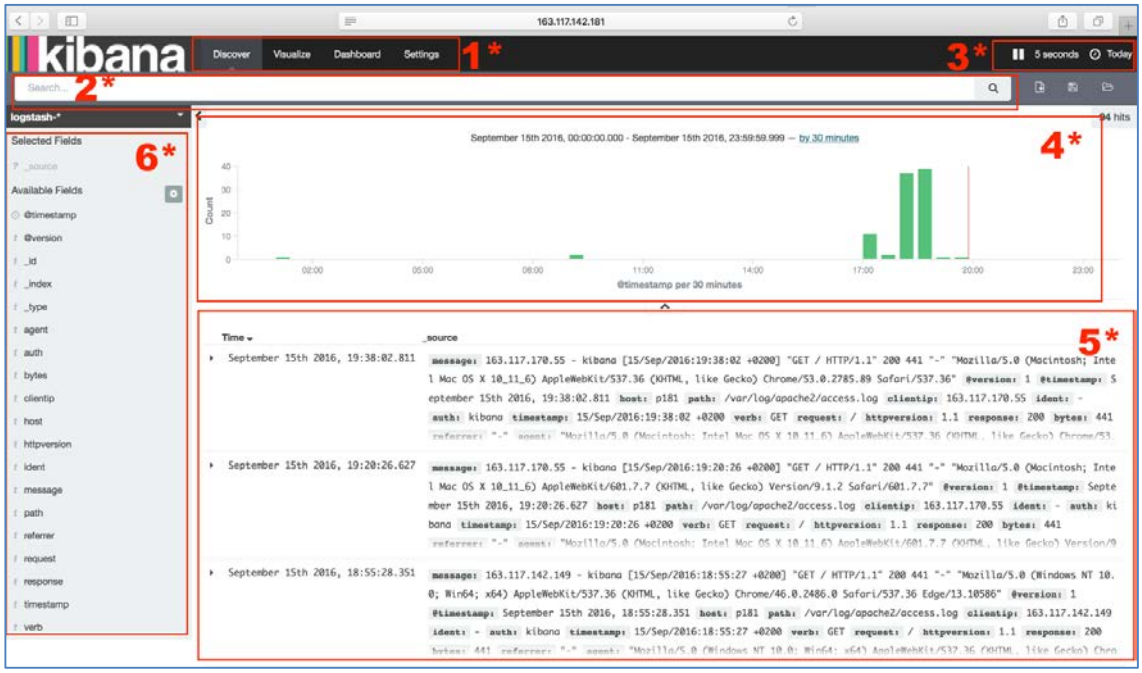


Ilustración 21: Pestaña Discover de Kibana.

donde podremos ver los siguientes elementos:

1. **Menú de opciones de Kibana:** menú principal de Kibana, permite acceder a las cuatro pestañas principales de Kibana:
    - **Discover:** permite llegar a esta misma ventana, incluye las opciones de búsqueda y recuperación de datos de Kibana.
    - **Visualize:** permite visualizar datos que han sido encontrados en la pestaña Discover, sus opciones se verán en la **ilustración 22.**
    - **Dashboard:** permite visualizar un conjunto de datos agrupados en un *dashboard*, sus opciones se verán en la **ilustración 23.**
    - **Settings:** permite revisar la configuración de Kibana, así como modificarla, sus opciones se verán en la **ilustración 24**.
2. **Barra de búsqueda de Kibana:** en esta barra de búsqueda podemos insertar las *querys* necesarias para poder visualizar datos de nuevas búsquedas, aquí podemos aplicar el caso de uso **CU-03.**
3. **Configuración de la visualización:** en esta pestaña podemos modificar el tiempo de auto refresco de los datos (entre cinco segundos y un día) y también podemos configurar el intervalo de la visualización (en fechas, horas y minutos).
4. **Visualización de datos:** gráfico interactivo de los datos recuperados por Kibana, es posible seleccionar intervalos de tiempo y visualizar información importante relativa a los datos haciendo *click* en ellos.

5. **Datos recuperados:** Lista completa de los datos que han sido recuperados por la *query* lanzada por Kibana.
6. **Lista de *Clave-valor*:** Lista de elementos *clave-valor*, indica que elementos son comunes en relación a los datos recuperados. Permite filtrar en función de los valores aquellos que coinciden con lo seleccionado.

Página de Visualización de datos:

En esta página podemos visualizar y modificar las búsquedas realizadas en la pestaña Discover. Las opciones incluyen cambiar el tipo de gráfico, modificar el eje de coordenadas, cambiar la métrica del eje de coordenadas, etc.

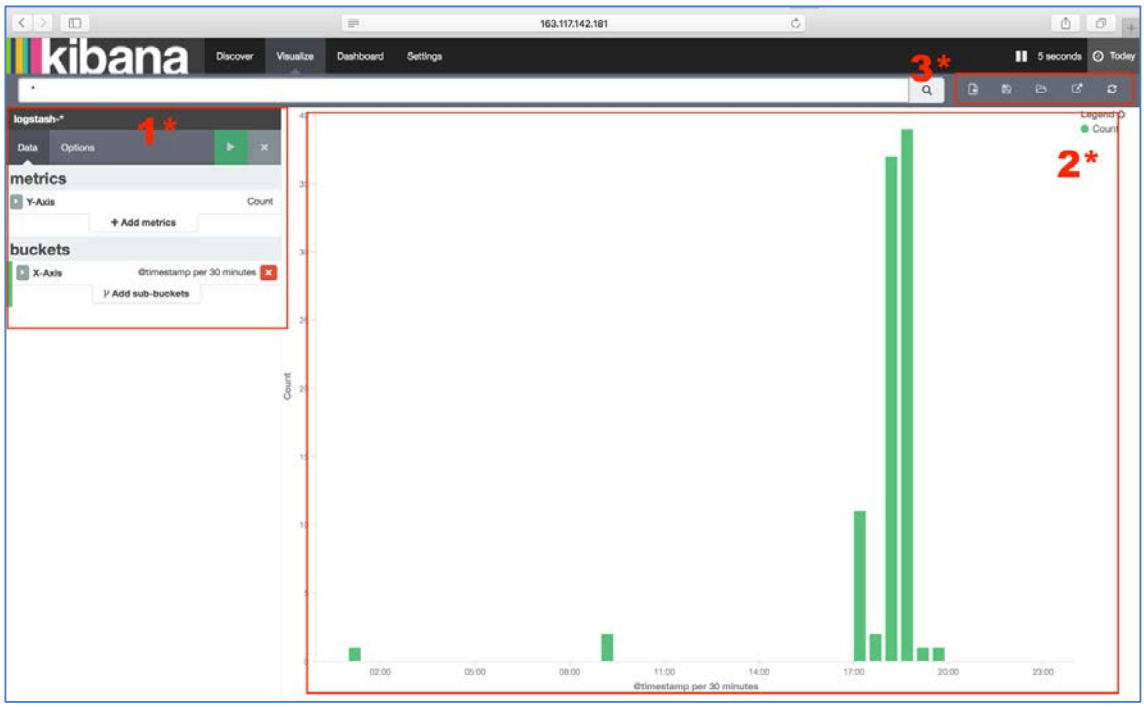


Ilustración 22: Pestaña Visualize de Kibana.

1. **Configuración de métricas del gráfico:** en esta lista podemos configurar, modificar y agregar los ejes de coordenadas del gráfico seleccionado.
2. **Gráfico de barras:** visualización de la vista actual del gráfico seleccionado.
3. **Barra de opciones del gráfico:** esta barra permite:
   - **Nueva visualización:** permite crear un gráfico nuevo en función de un conjunto de datos seleccionados en la pestaña anterior.
   - **Grabar visualización:** permite agregar un gráfico nuevo, aquí podemos aplicar el caso de uso **CU-04.**
   - **Cargar una visualización almacenada:** permite abrir y visualizar un gráfico guardado anteriormente.
   - **Compartir una visualización:** genera código para compartir el gráfico.
   - **Actualizar la visualización actual:** permite refrescar el gráfico actual.

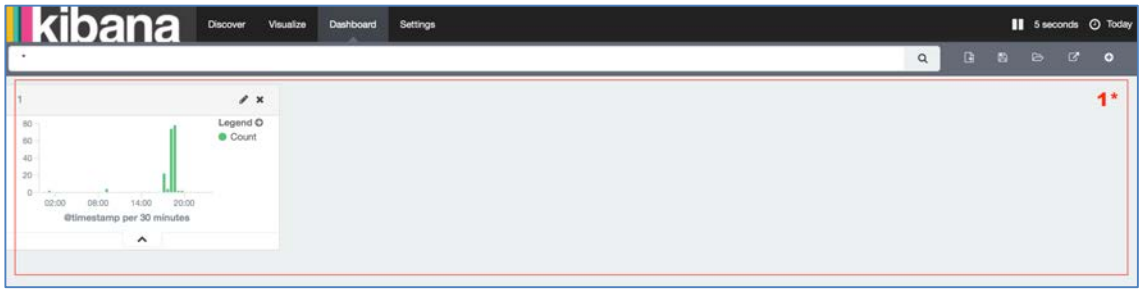


Ilustración 23: Pestaña Dashboard de Kibana

1. **Escritorio de gráficos o *Dashboard*:** permite visualizar e integrar nuevos gráficos, eliminar gráficos existentes e integrar en un solo espacio de trabajo todos los gráficos necesarios. El espacio de trabajo es ampliamente configurable e intuitivo, también es posible agregar, almacenar y exportar *dashboards* dentro de este entorno de trabajo.

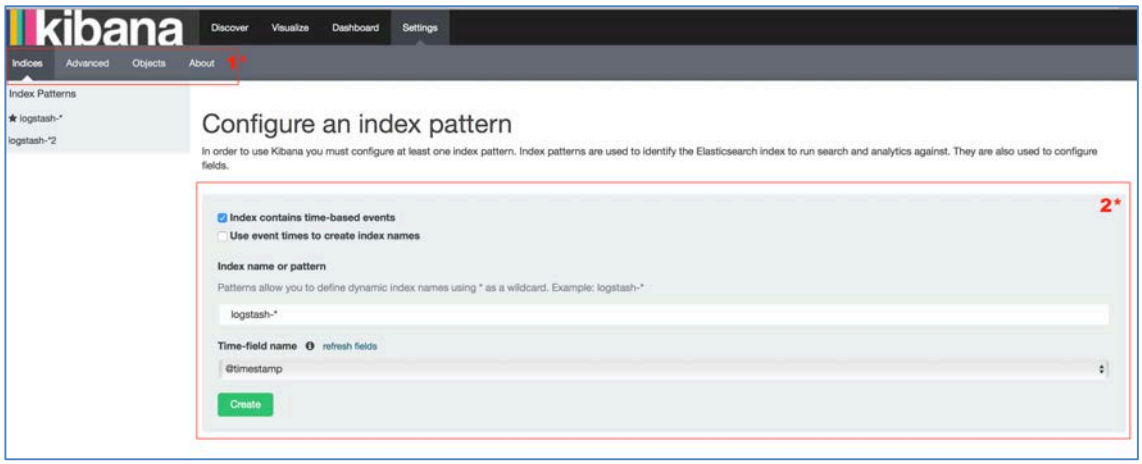


Ilustración 24: Pestaña Settings de Kibana.

2. **Sub-Menú de opciones de configuración:** en este sub-menú encontramos las siguientes opciones:
    - **Indices:** permite llegar a esta vista, incluye las opciones de creación y configuración de un patrón indexado.
    - **Advanced:** lleva a la vista Advanced, sus opciones se verán en la **ilustración 25.**
    - **Objects:** lleva a la vista Objects, sus opciones se verán en la **ilustración 26.**
    - **About:** lleva a la vista About, sus opciones se verán en la **ilustración 27.**
3. **Configuración de un patrón indexado:** aquí es posible crear y configurar un patrón indexado, esto permite conectar con Elasticsearch en una dirección específica.

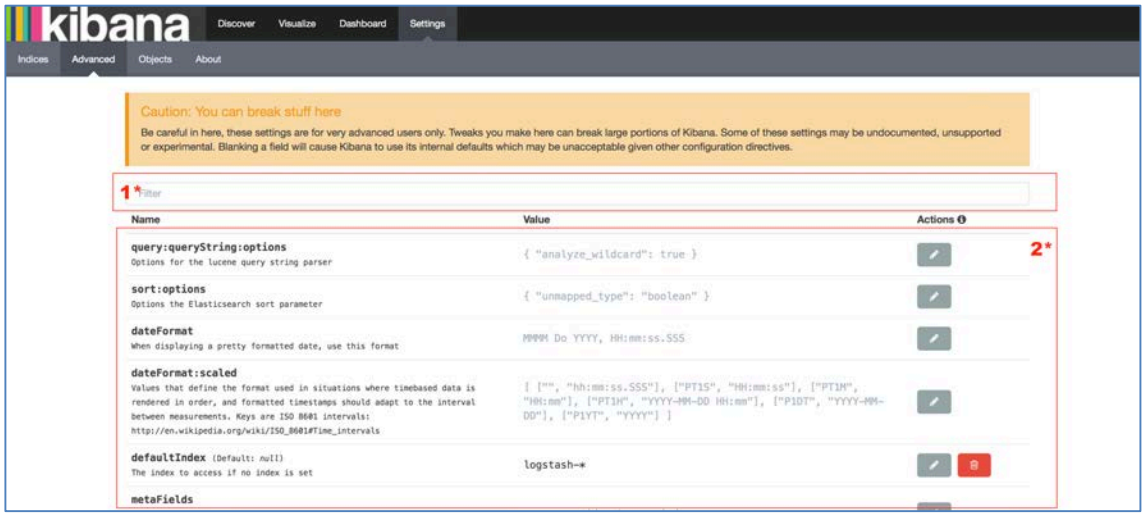


Ilustración 25: Sub-Menú Advanced.

1. **Filtrado de configuraciones de Kibana:** permite filtrar las opciones de Kibana que coincidan con la búsqueda, de esta forma es más rápido acceder a las configuraciones y así no es necesario buscar exhaustivamente entre todas las opciones.
2. **Configuración de Kibana:** permite entrar a las configuraciones de Kibana, estas opciones son avanzadas y no es recomendable cambiarlas.

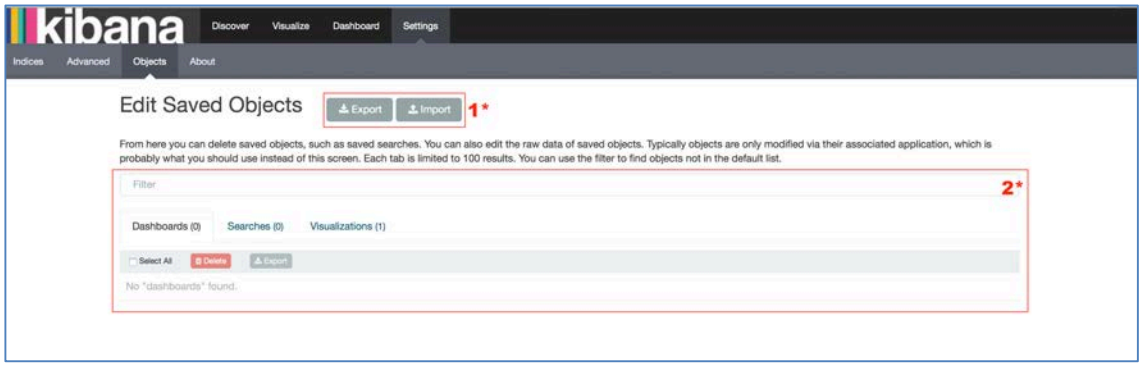


Ilustración 26: Sub-Menú Objects

1. **Exportar e importar Objetos:** permite importar y exportar las configuraciones de Kibana en ficheros en formato *JSON*.
2. **Objetos importados en Kibana:** permite visualizar aquellos objetos que han sido importados dentro de la configuración de Kibana.

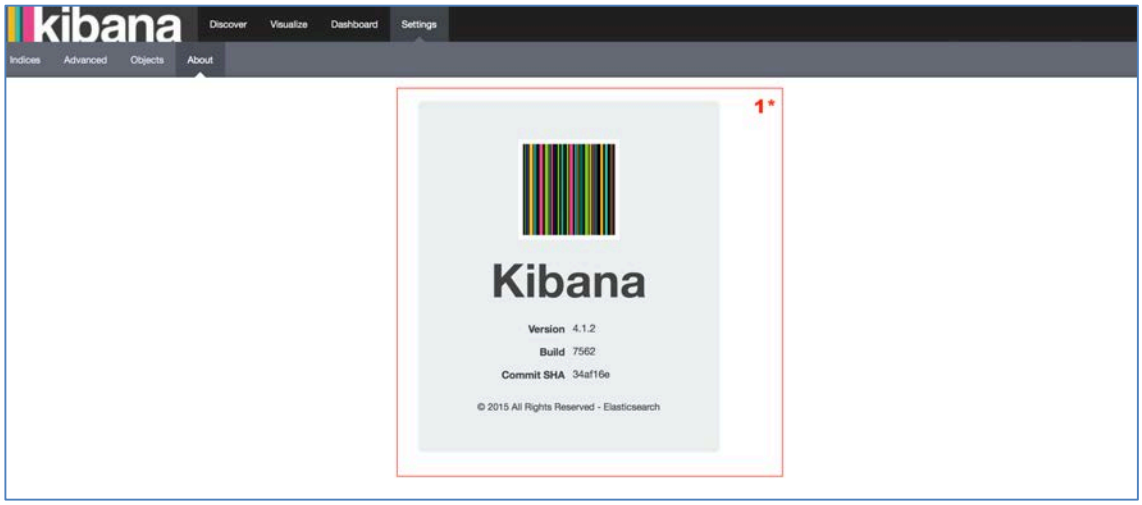


Ilustración 27: Sub-Menú About.

1. **Acerca de, versión de Kibana:** vista de Kibana que permite visualizar la información de la versión del software, la versión de la compilación y la comprobación de la autenticidad de la misma.

5. Implantación

En este apartado revisaremos cuáles son los pre-requisitos y los pasos necesarios para la implantación del sistema en el laboratorio del Departamento de Informática. Los detalles más profundos serán vistos en el **Apéndice I – Manual de instalación y configuración** que se encuentra en el **Capítulo Apéndices** de este documento.

5.1. Pre-requisitos

Antes de continuar es necesario identificar cuáles son los requisitos que deben cumplirse previamente a la instalación del sistema. Revisaremos detalladamente cada uno de ellos, indicando el comando necesario para su ejecución cuando corresponda.

Es importante destacar que la instalación de los pre-requisitos no es parte de los objetivos de este documento, por lo cual es responsabilidad del equipo del laboratorio del Departamento de Informática disponer de esta parte imprescindible para la implantación antes de empezar.

La lista de requisitos es la siguiente:

- Instalación del *S.O. Debian* versión "Jessie" 8.5, instalación estándar (19).
- Configuración de la red *LAN* y *servidores DNS* del laboratorio.
- Instalación del software de *servidor SSH*, instalación por medio de línea de comandos.
    - apt-get install ssh
- Comando sudo, instalación por medio de línea de comandos.
    - apt-get install sudo
- instalación del *servidor* web Apache 2 según el tutorial de ite.educacion.es (20).
- Ejecutar apt-get upgrade y apt-get update.
- Creación de una cuenta sin privilegios de súper usuario en el ordenador *nodo* maestro de Elasticsearch.
    - useradd –p elastic nMaestro –g nobody
- Apertura de puertos necesarios para la correcta ejecución del sistema en el *nodo* maestro:
    - Iptables -I INPUT -p all -j DROP
    - iptables -I INPUT -p tcp --dport 23 -j ACCEPT
    - iptables -I INPUT -p tcp --dport 80 -j ACCEPT
    - iptables -I INPUT -p tcp --dport 443 -j ACCEPT
    - iptables -I INPUT -p tcp --dport 5601 -j ACCEPT
- Incluir el *script* de la configuración de *iptables* en el arranque del sistema (21).
- Instalación de JRE de Java en su versión 8 (22).

5.2. Configuración del entorno operacional

En este apartado revisaremos la configuración del entorno donde se implantará la solución. Básicamente, es un extracto del **Apéndice I – Manual de instalación y configuración** del **Capítulo Apéndices** que nos permitirá entender cuáles son las configuraciones necesarias para la correcta implantación de la solución.

En el *nodo* maestro, implantaremos dos de las tres capas de la arquitectura: la capa de indexación y la capa de visualización. Esto implica que Kibana y Elasticsearch serán implantados en este ordenador. La dirección IP de a configurar para el *nodo* maestro es la que ya se ha indicado: **163.117.142.181.**

Primero, procederemos a instalar Elasticsearch, para ello dentro de un *terminal* en la línea de comando debemos ejecutar lo siguiente:

```
1 wget -qO - https://packages.elastic.co/GPG-KEY-elasticsearch | sudo apt-key add -
2 echo "deb http://packages.elastic.co/elasticsearch/1.7/debian stable main" | sudo tee –a /etc/apt/sources.list.d/elasticsearch-1.7.list
3 sudo apt-get update
4 sudo apt-get install elasticsearch
```

Con esto agregaremos el repositorio de Elasticsearch para apt para luego instalarlo. El siguiente paso es incluir la siguiente información el fichero de configuración en la ruta: **/etc/elasticsearch/elasticsearch.yaml**

```
1 cluster.name: elasticnode
2 discovery.zen.ping.multicast.enabled: false
3 index.number_of_replicas: 0
```

Luego de esto, procederemos a instalar Kibana, para esto, debemos ingresar los siguientes comandos.

```
1 cd /opt
2 sudo wget https://download.elastic.co/kibana/kibana/kibana-4.1.2-linux-x64.tar.gz
3 sudo tar xvzf kibana-4.1.2-linux-x64.tar.gz
```

Luego, en la ruta **/etc/systemd/system/kibana.service** debemos crear un fichero con la siguiente información dentro.

```
1 [Unit]
2 Description=open source browser based analytics and search dashboard for Elasticsearch
3
4 [Service]
5 Type=simple
6 ExecStart=/opt/kibana-4.1.2-linux-x64/bin/kibana -c /opt/kibana-4.1.2-linux-x64/config/kibana.yml
```

Después, debemos iniciar los servicios Kibana, Elasticsearch y Apache.

```
sudo systemctl start apache2
sudo systemctl start elasticsearch
sudo systemctl start kibana
```

Con esto ya tenemos el *nodo* maestro funcionando, el siguiente paso es configurar los ordenadores de las aulas para que envíen información al *nodo* maestro. Para esto debemos ejecutar las siguientes líneas de comando en una *terminal* en cada uno de ellos.

```
wget -qO - https://packages.elasticsearch.org/GPG-KEY-elasticsearch | sudo apt-key add -
echo "deb http://packages.elasticsearch.org/logstash/1.5/debian stable main" | sudo tee -a /etc/apt/sources.list
sudo apt-get update
sudo apt-get install logstash
```

Con esto tenemos Logstash instalado, lo siguiente es configurar los filtros para hacer que haga su trabajo, un ejemplo de un filtro es:

```
input {
  file {
    path => '/var/log/apache2/access.log'
  }
}

filter {
  grok {
    match => { "message" => "%{COMBINEDAPACHELOG}" }
  }
}

output {
  stdout { codec => rubydebug }
}
```

Donde el parámetro “path” indica la ruta del *log* a recuperar, el parámetro “match” indica que tipo de *log* se va a recuperar y el output indica que la salida es la estándar. Todos los parámetros son configurables siguiendo la estructura de la configuración de un filtro de Logstash.

La idea principal es que Logstash recupera la información de entrada que está señalada, la analiza o filtra para luego enviarla a una salida determinada. Cada uno de estos apartados representa las respectivas partes de este proceso.

Luego de crear el filtro, es necesario poner el archivo en la siguiente ruta: **/etc/logstash/conf.d/nombre_del_filtro.conf**, por último, es necesario informar a Logstash de que debe actualizar su lista de filtros.

```
sudo /opt/logstash/bin/logstash -f /etc/logstash/conf.d
```

Para más información sobre la creación de filtros, el lector puede recurrir a la página oficial de Logstash (21) o a una web de tutorial de creación de *logs* (22). Para seguir un tutorial sobre como configurar ELK, puede recurrir a linode.com (23).

Finalmente, debemos configurar la página inicial de la aplicación en el *nodo* maestro de tal forma que deshabilite el acceso no autorizado, para ello, utilizaremos la configuración htaccess de Apache.

Primero, cambiaremos la información de la página de inicio de Apache para que redirija automáticamente a la página de Kibana.

```
 GNU nano 2.2.6                Fichero: index.html

<!DOCTYPE HTML>
<html>
  <head>
    <meta http-equiv="refresh" content="0; url=http://163.117.142.181:5601/" />
  </head>
  <body>
  </body>
</html>
```

Ilustración 28: Nuevo fichero index.html

Este fichero no contiene información, pero por medio del comando “refresh” re direcciona automáticamente a la *URL* especificada en el campo “url”. Lo siguiente es configurar un fichero .htpasswd y un fichero .htaccess. Para el primer fichero debemos crear ambos ficheros en blanco con el siguiente comando:

```
1  touch /var/www/html/.htpasswd
2  touch /var/www/html/.htaccess
```

La recuperación de estos ficheros por medio de http desde el exterior esta desactivada por defecto en Apache desde la versión 2, lo cual nos permite almacenar las contraseñas de forma segura. Luego, debemos editar el fichero .htaccess agregando la siguiente información

```
1  AuthName "Se necesita un nombre de usuario y una contraseña "
2  AuthType Basic
3  require valid-user
4  AuthUserFile /var/www/html
```

Finalmente, debemos crear un usuario para que pueda iniciar sesión en el sistema, los datos para ello serán: **Usuario:** “kibana”, **Contraseña:** “articuno”

El comando para configurar los datos de acceso es:

```
1  htpasswd /var/www/html kibana
```

Apache solicitara ingresar la nueva contraseña para el usuario Kibana y confirmarla, es necesario escribir la contraseña en la *terminal* exactamente igual ambas veces. Finalmente, el último paso es generar el alias del fichero htaccess para hacer que la configuración funcione, para esto debemos editar el fichero **default.conf** dentro de la ruta **/etc/apache2/sites-avalibles/** agregando la información que se muestra en la siguiente imagen.

```
 GNU nano 2.2.6                          Fichero: /etc/apache2/sites-available/000-default.conf

<VirtualHost *:80>
        # The ServerName directive sets the request scheme, hostname and port that
        # the server uses to identify itself. This is used when creating
        # redirection URLs. In the context of virtual hosts, the ServerName
        # specifies what hostname must appear in the request's Host: header to
        # match this virtual host. For the default virtual host (this file) this
        # value is not decisive as it is used as a last resort host regardless.
        # However, you must set it for any further virtual host explicitly.
        #ServerName www.example.com

        ServerAdmin webmaster@localhost
        DocumentRoot /var/www/html

        Alias /html /var/www/html
        <Directory "/var/www/html">
          Options Includes
          AllowOverride All
          Order allow,deny
          Allow from all
        </Directory>


        # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
        # error, crit, alert, emerg.
        # It is also possible to configure the loglevel for particular
        # modules, e.g.
        #LogLevel info ssl:warn

        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log combined

        # For most configuration files from conf-available/, which are
        # enabled or disabled at a global level, it is possible to
        # include a line for only one particular virtual host. For example the
        # following line enables the CGI configuration for this host only
        # after it has been globally disabled with "a2disconf".
        #Include conf-available/serve-cgi-bin.conf
</VirtualHost>
```

Ilustración 29: Nuevo fichero default.conf

Los apartados que debemos agregar son la línea Alias y las seis líneas siguientes. Para más información de cómo configurar Apache para el uso de htaccess se recomienda al lector revisar el tutorial de la web www.alcancelibre.org (24).

La ultima configuración necesaria de Apache es la inclusión del protocolo SSL para la conexión segura con el sitio *web*. La configuración estándar de Apache se puede revisar en el tutorial de la *web* de digicert.com (25). Los pasos para configurar el certificado SSL auto firmado son los siguientes:

```
1  openssl genrsa -out ca.key 1024
2  openssl req -new -key ca.key -out ca.csr
3  openssl x509 -req -days 365 -in ca.csr -signkey ca.key -out ca.crt
```

Luego de crear y firmar los certificados, es necesario moverlos a una ruta donde puedan ser accedidos por Apache, para ello:

```
1  mv ca.crt /etc/pki/tls/certs
2  mv ca.key /etc/pki/tls/private/ca.key
3  mv ca.csr /etc/pki/tls/private/ca.csr
```

Después de eso, debemos forzar a que Apache use los certificados, modificando las líneas del fichero **/etc/httpd/conf/ssl.conf**

```
SSLCertificateFile /etc/pki/tls/certs/ca.crt
SSLCertificateKeyFile /etc/pki/tls/private/ca.key
```

Para luego configurar el fichero que permite la conexión segura, este fichero es el siguiente: **/etc/httpd/conf/https.conf**, y las líneas a agregar son:

```
NameVirtualHost *:443

<VirtualHost *:443>
    SSLEngine on
    SSLCertificateFile /etc/pki/tls/certs/ca.crt
    SSLCertificateKeyFile /etc/pki/tls/private/ca.key

    AllowOverride All

    DocumentRoot /var/www/httpsdocs
    ServerName nombresitio.com
</VirtualHost>
```

Con esto último creamos un nuevo huésped virtual, que nos permite iniciar comunicaciones seguras. Para asegurarnos de que la configuración funciona perfectamente debemos reiniciar el *daemon* de Apache:

```
sudo systemctl restart apache2
```

Y en los ordenadores de las aulas debemos reiniciar el *daemon* de Logstash con el siguiente comando:

```
sudo systemctl restart logstash
```

Haciendo esto habremos finalizado la configuración del sistema ELK implantado dentro de las aulas informáticas del laboratorio del Departamento de Informática. La configuración restante es en sí misma la personalización que los técnicos del laboratorio deseen ejecutar, entiéndase el *dashboard* de Kibana y los filtros necesarios para la recuperación de los *logs* que estimen convenientes.

6. Pruebas

Este capítulo nos adentrará en el plan de pruebas del proyecto. Básicamente, con él perseguimos revisar el funcionamiento correcto del desarrollo del proyecto y asegurar el buen funcionamiento del mismo basado en el cumplimiento de los requisitos del usuario.

6.1. Especificación del plan de pruebas

En este apartado revisaremos los tipos de pruebas que serán realizados en el desarrollo del plan de calidad de este proyecto. Es importante destacar que durante éste capítulo revisaremos puntualmente tres tipos de pruebas:

- **Pruebas de implantación:** verifican que el proceso de implantación cumple el funcionamiento especificado de la aplicación, permitiendo su uso en el entorno del laboratorio del Departamento de Informática.

- **Pruebas de validación:** verifican que la aplicación cumple las funciones del sistema que reemplazará y además cumple con las funciones prometidas.

- **Pruebas de cumplimiento de restricciones**: verifican el cumplimiento de las restricciones impuestas por los requisitos de restricción del **Capítulo 3**.

En general, cada una de las pruebas permiten corroborar el cumplimiento de este proyecto, identificando el cumplimiento o no cumplimiento de los requisitos necesarios para que el sistema sea aprobado.

6.2. Especificación técnica del plan de pruebas

En este apartado revisaremos las pruebas realizadas al sistema. Para poder estandarizar el entendimiento de las mismas utilizaremos una plantilla que nos permitirá identificar fácilmente la información relevante relacionada con la prueba realizada.

Antes de proceder a ver la plantilla, revisaremos los apartados relacionados con la misma. Esto nos permitirá entender a que corresponde cada uno y que información contendrá:

- **Identificador:** permite diferenciar unívocamente la prueba. Tendrá el formato **PY-NN**, donde el formato indica lo siguiente:
  P: indica que la tabla corresponde a una prueba del sistema
  Y: indica el tipo de prueba del sistema: **I** para implantación, **V** para validación y **C** para cumplimiento.
  NN: indica el numero correlativo de la prueba que es unívoco.

- **Objetivos:** indica la finalidad de la prueba, el fin de la realización de la misma.
- **Necesidades:** prerrequisitos que son necesarios para la realización de la prueba.
- **Entradas:** valores de entrada necesarios para la realización de la prueba.
- **Secuencia:** pasos necesarios para que la prueba pueda realizarse.
- **Salidas:** valores de salida o resultado de la realización de la prueba.
- **Requisitos relacionados:** requisitos que se corresponden con la prueba y que son contrastados al realizar la misma.

Luego, la matriz que se muestra a continuación está compuesta de todos estos elementos y es en si la plantilla que utilizaremos para cada una de las pruebas.

Identificador	
Objetivos	
Necesidades	
Entradas	
Secuencia	
Salidas	
Requisitos relacionados	

Tabla 31: Plantilla de pruebas.

Finalmente, las pruebas realizadas al sistema son las siguientes:

PV-01	
Objetivos	Verificar que el sistema es capaz de leer *logs* de los ordenadores de las aulas.
Necesidades	El componente Logstash debe estar instalado en las aulas informáticas.
Entradas	*Logs* de información del uso de recursos.
Secuencia	Logstash recupera la información a medida que se va generando
Salidas	La información recuperada en formato *JSON*.
Requisitos relacionados	RC-01

Tabla 32: Plantilla de pruebas.

PV-02	
Objetivos	Verificar que el sistema es capaz de recuperar los términos de los *logs* por separado.
Necesidades	El componente Logstash debe estar instalado en las aulas informáticas.
Entradas	*Logs* de información del uso de recursos.
Secuencia	Logstash recupera la información a medida que se va generando, la separa y la entrega en formato *clave-valor* en un documento *JSON*
Salidas	La información recuperada en formato *JSON*.
Requisitos relacionados	RC-02

Tabla 33: Plantilla de pruebas.

PV-03	
Objetivos	Verificar que el sistema es capaz de recuperar el formato de tiempos de los *logs* y estandarizarlos.
Necesidades	El componente Logstash debe estar instalado en las aulas informáticas.
Entradas	Los datos recuperados previamente en distintos formatos
Secuencia	Logstash recupera los datos en distintos formatos y los estructura en un formato común, reconociéndolos por medio de expresiones regulares y entregándola información en un formato estándar.
Salidas	Información del tiempo estandarizada en un formato.
Requisitos relacionados	RC-03

Tabla 34: Plantilla de pruebas.

PV-04	
Objetivos	Verificar que el sistema almacena la información en una base de datos.
Necesidades	El sistema debe estar implantado.
Entradas	-
Secuencia	Elasticsearch es la base de datos *NoSQL* que forma parte del sistema, para ser precisos, es un sistema de indexado distribuido y consultable.
Salidas	-
Requisitos relacionados	RC-04

Tabla 35: Plantilla de pruebas.

PV-05	
Objetivos	Verificar que el sistema cuenta con una interfaz gráfica.
Necesidades	El sistema debe estar implantado.
Entradas	-
Secuencia	Kibana es la interfaz gráfica del sistema, permite utilizar las opciones más comunes del mismo y consultar la base de datos.
Salidas	-
Requisitos relacionados	RC-05

Tabla 36: Plantilla de pruebas.

PV-06	
Objetivos	Verificar que el sistema muestra el avance de del uso de recursos en función del tiempo.
Necesidades	El sistema debe estar implantado, debe haber una búsqueda con su grafico correspondiente almacenados en el sistema.
Entradas	Información en semi-tiempo real sobre el uso de recursos
Secuencia	El sistema recupera la información y muestra los detalles en una gráfica bidimensional.
Salidas	La grafica avanza en función del tiempo y de los parámetros configurados en la misma.
Requisitos relacionados	RC-06

Tabla 37: Plantilla de pruebas.

PV-07	
Objetivos	Verificar que el sistema permite identificar fallas y caídas de servicios.
Necesidades	El sistema debe estar implantado, debe haber una búsqueda con su grafico correspondiente almacenados en el sistema.
Entradas	Discontinuidad de la información recibida en semi-tiempo real mientras el sistema funciona.
Secuencia	El sistema muestra un cese de visualización de información, los técnicos de laboratorio pueden detectar la caída de un servicio.
Salidas	Visualización de un cese de recepción de información en la gráfica.
Requisitos relacionados	RC-07

Tabla 38: Plantilla de pruebas.

PV-08	
Objetivos	Verificar que el sistema permite recuperar información de la base de datos desde la interfaz gráfica.
Necesidades	El sistema debe estar implantado
Entradas	-
Secuencia	El sistema permite hacer *querys* y recuperar información avanzada por medio de Kibana.
Salidas	Información avanzada relacionada con los datos recuperados de la *query*.
Requisitos relacionados	RC-08

Tabla 39: Plantilla de pruebas.

PV-09	
Objetivos	Verificar que el sistema se accesible desde fuera de la red de la Universidad.
Necesidades	El sistema debe estar implantado, el *nodo* maestro debe contener una dirección IP publica, el cliente debe tener acceso a internet.
Entradas	Conexión remota desde un *navegador web* desde el ordenador del cliente.
Secuencia	El cliente se conecta por medio de un *navegador web* a Kibana en la IP 163.117.142.181, debe visualizar la pantalla de inicio de sesión.
Salidas	Kibana debe mostrarse por medio del *navegador web*.
Requisitos relacionados	RC-09

Tabla 40: Plantilla de pruebas.

PV-10	
Objetivos	Verificar que el sistema se accesible todo el año.
Necesidades	El sistema debe estar implantado.
Entradas	-
Secuencia	Mientras el sistema se mantenga bajo el soporte de los técnicos del laboratorio de informática, este debería poder cumplir con su funcionamiento de forma continua.
Salidas	Funcionamiento continuo
Requisitos relacionados	RC-10

Tabla 41: Plantilla de pruebas.

PV-11	
Objetivos	Verificar que el sistema permite incluir *plugins*.
Necesidades	El sistema debe estar implantado.
Entradas	El sistema adquiere un nuevo filtro
Secuencia	El componente Logstash adquiere un nuevo filtro de recursos y lo utiliza.
Salidas	El nuevo filtro genera nuevos resultados de la información que recoge de los ordenadores de las aulas.
Requisitos relacionados	RC-11

Tabla 42: Plantilla de pruebas.

PI-12	
Objetivos	Verificar que la interfaz gráfica sea accesible solo por medio de usuario y contraseña
Necesidades	El sistema debe estar implantado, el acceso al sistema debe estar configurado.
Entradas	Información de nombre de usuario y contraseña
Secuencia	Los datos obtenidos se procesan y se comparan con los datos de usuario y contraseña de conexión al sistema, si no coinciden, no se procede al acceso a Kibana, si coinciden, el usuario visualizará el componente Kibana.
Salidas	Acceso al sistema, Página de error que informa que los datos introducidos son erróneos.
Requisitos relacionados	RR-01

Tabla 43: Plantilla de pruebas.

PC-13	
Objetivos	Verificar que el sistema sea *horizontalmente escalable*.
Necesidades	Identificar las especificaciones técnicas del sistema.
Entradas	-
Secuencia	El sistema implantado cumple con el requisito, según la documentación de mismo, es posible agregar *nodos* al sistema de indexación, por lo cual es *horizontalmente escalable*.
Salidas	-
Requisitos relacionados	RR-02

Tabla 44: Plantilla de pruebas.

PC-14	
Objetivos	Verificar que la base de datos de sistema es consultable.
Necesidades	Identificar las especificaciones técnicas del sistema.
Entradas	-
Secuencia	Según las especificaciones del producto, es posible hacer *querys* contra la base de datos Elasticseach. Esta funcionalidad ha sido comprobada.
Salidas	-
Requisitos relacionados	RR-03

Tabla 45: Plantilla de pruebas.

PC-15	
Objetivos	Verificar que el sistema sea *open source*.
Necesidades	Identificar las especificaciones técnicas del sistema.
Entradas	-
Secuencia	Según las especificaciones del producto, el sistema pertenece a una licencia Apache 2, lo cual lo convierte en un producto *open source.*
Salidas	-
Requisitos relacionados	RR-04

Tabla 46: Plantilla de pruebas.

PI-16	
Objetivos	Verificar que el sistema sea compatible con los recursos y la arquitectura actual de los equipos de la Universidad.
Necesidades	El sistema debe estar implantado.
Entradas	Prerrequisitos de implantación
Secuencia	El sistema debe poder funcionar dentro de una red LAN, debe poder trabajar dentro del entorno del laboratorio del departamento de informática.
Salidas	Luego de confirmar que el sistema cumple los prerrequisitos, es viable la implantación y por lo tanto puede procederse a instalar.
Requisitos relacionados	RR-05

Tabla 47: Plantilla de pruebas.

PC-17	
Objetivos	Verificar que el sistema se comunique de forma segura con los clientes por medio del protocolo *SSL*.
Necesidades	El sistema debe estar implantado, el navegador del cliente debe soportar HTTPS, la conexión con el sistema mediante la red debe estar disponible.
Entradas	
Secuencia	
Salidas	
Requisitos relacionados	RR-06

Tabla 48: Plantilla de pruebas.

6.3. Matrices de trazabilidad

Finalmente, en la matriz que podemos ver a continuación, podemos comprobar que cada requisito especificado en el **Capítulo 3** de este documento se corresponde con una prueba de las que se han realizado en el apartado anterior. Esto permite confirmar al lector que se ha corroborado el buen funcionamiento del sistema.

Prueba/ requisito	RC-01	RC-02	RC-03	RC-04	RC-05	RC-06	RC-07	RC-08	RC-09	RC-10	RC-11	RR-01	RR-02	RR-03	RR-04	RR-05	RR-06
PV-01	X																
PV-02		X															
PV-03			X														
PV-04				X													
PV-05					X												
PV-06						X											
PV-07							X										
PV-08								X									
PV-09									X								
PV-10										X							
PV-11											X						
PI-12												X					
PC-13													X				
PC-14														X			
PC-15															X		
PI-16																X	
PC-17																	X

Tabla 49: Matriz de trazabilidad de pruebas.

7. Planificación y presupuesto

En este apartado revisaremos la perspectiva temporal y económica de este proyecto. La idea es poder exponer la programación estimada de las fases del mismo tomando en cuenta el cálculo detallado de los costes y beneficios obtenidos.

7.1. Planificación

La programación inicial de este proyecto se estimó en función de la asignatura Trabajo de fin de grado, a la cual se le asignan **12 créditos ECTS**. Estos créditos corresponden a un total de **360hrs** de las cuáles hemos descontado un 10% previsto para ajustes finales y margen de error de tiempo.

La fecha final del proyecto es la fecha de entrega prevista de los Trabajos de final de grado en su convocatoria extraordinaria que corresponde al periodo comprendido entre los días 20 al 26 de septiembre del año 2016.

Calculando una carga de **13,5hrs de trabajo más 3hrs de documentación a la semana** podemos adjudicar una **duración total de 20 semanas**. Con esto se ha determinado que la fecha de inicio de este proyecto será el día **lunes 2 de mayo del año 2016**.

Las fases que se desarrollaran a lo largo de este proyecto son las que se muestran a continuación:

- **Documentación.**
- **Propuesta de proyecto.**
  - Toma de requisitos.
  - Validación de los requisitos con el cliente.
- **Estudio de mercado.**
- **Análisis.**
  - Análisis de los requisitos.
  - Trazabilidad entre soluciones y requisitos.
  - Elección de la solución y validación con el cliente.
- **Diseño.**
  - Diseño de la Arquitectura.
  - Diseño de Casos de uso.
- **Implantación.**
  - Verificación de Pre-requisitos
  - Implantación del sistema en las aulas informáticas.
  - Implantación del *nodo* maestro de la solución.
- **Evaluación.**
  - Plan de pruebas.
  - Evaluación de resultados.

Viendo la siguiente carta Gantt, podemos hacer un seguimiento cronológico de lo ocurrido en el desarrollo del proyecto.

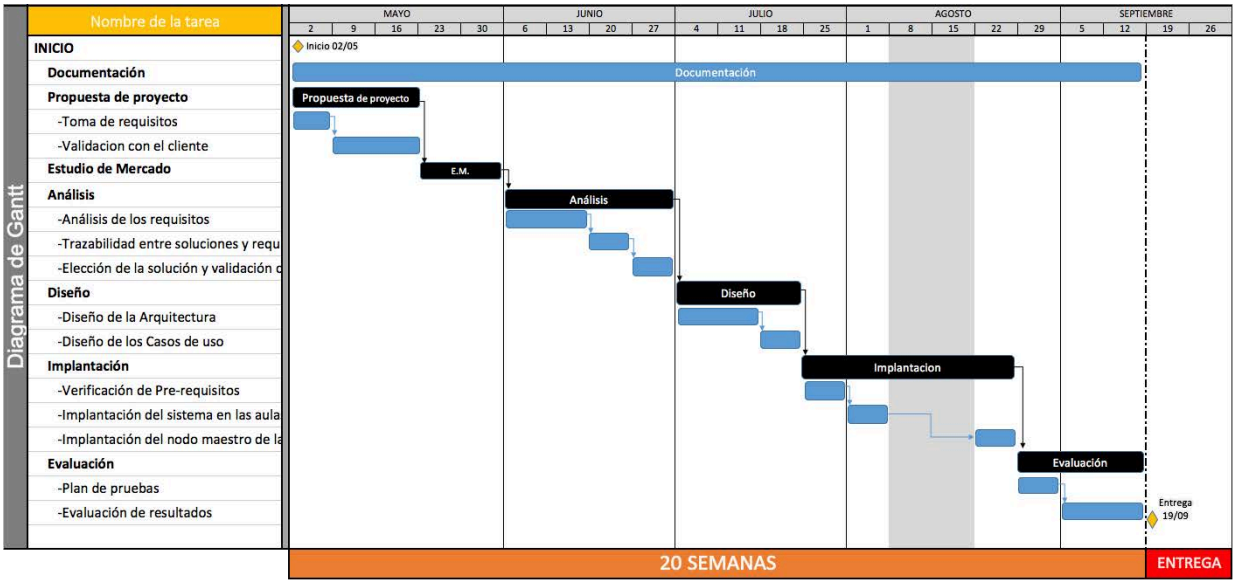


Ilustración 30: Carta Gantt de la planificación del proyecto.

En el diagrama de Gantt de la planificación podemos visualizar cual es la planificación del proyecto, podemos ver desde el día ocho de agosto hasta el día veintiuno de agosto la Universidad permanecerá cerrada, por lo cual la implantación se verá interrumpida. Para visualizar mayores detalles, en el **Capítulo Apéndices – Apéndice III: Detalles de la planificación, diagrama de Gantt** es posible ver una vista horizontal de la **ilustración 20**.

7.2. Presupuesto

Basados en lo antes visto, podemos tener una idea del coste en horas hombre del proyecto, pero debemos traducir ese resultado a números. Para ello debemos identificar los costes asociados y calcular la suma total de ellos, luego debemos añadir el margen de riesgo y finalmente el margen de beneficio, esto nos entregara el precio final del proyecto sin impuestos.

Debemos tomar en cuenta que el desglose real de los costes asociados se divide en tres grupos:

- **Costes del personal:** corresponde al coste de las horas hombre que se necesita para desarrollar el proyecto.

- **Costes de los equipos:** corresponde al coste asociado a las herramientas y elementos necesarios para desarrollar el proyecto.

- **Costes indirectos:** corresponde a los costes asociados al entorno donde se desarrolla el proyecto, costes relacionados con los gastos de facturas comunes a este proyecto y a otros que se hagan en paralelo.

Luego de esta aclaración, revisaremos a cuánto ascienden estos costes:

Costes del personal

Salarios			
Salario bruto anual			**26.000,00 €**
Salario bruto mensual			**2.166,67 €**
Cuota patronal			
Seguridad Social	23,6	%	**511,33 €**
Desempleo	5,5	%	**119,17 €**
Formación profesional	0,6	%	**13,00 €**
Fogasa	0,2	%	**4,33 €**
Total			**647,83 €**
Planificación del proyecto			
Duración del proyecto en horas			**330**
Duración del proyecto en semanas			**20**
Horas de trabajo a la semana			**16,5**
Horas de trabajo al mes			**66**
Coste total del personal			
Coste mensual del trabajador			**2.814,50 €**
Coste de la hora trabajada			**42,64 €**
Total de Coste de horas hombre del proyecto			**14.072,50 €**

Tabla 50: Coste de las horas del personal.

Costes de los Equipos

Coste de los equipos	
Servidor Maestro ELK – I5 3,5GHZ/4GB/320GB	**367,80 €**
Licencia ELK	**0,00 €**
Total de Costes de los Equipos	**367,80 €**

Tabla 51: Coste de los equipos del proyecto.

Costes indirectos

Equipos de desarrollo y documentación				
Apple Macbook Pro Retina 15' i7 2,4GHZ/8GB/250GB				**2.313,22 €**
Monitor Samsung SyncMaster EX1920				**134,47 €**
Impresora HP Laserjet 1018n				**147,92 €**
Amortización de equipos de desarrollo y documentación				
concepto	**amort. anual**	**amort. semanal**	**semanas**	**total**
Apple Macbook Pro Retina 15' i7/8GB/250GB	1.156,61 €	22,26	20	**445,20 €**
Monitor Samsung SyncMaster EX1920	67,24 €	1,3	20	**26,00 €**
Impresora HP Laserjet 1018n	73,96 €	1,42	20	**28,40 €**
Total				**499,60 €**
Otros costes				
Alquiler del local(luz y agua incluidos)				**550,00 €**
Consumibles ofimática				
Total				**550,00 €**
Total Costes indirectos				
Total de Costes indirectos				**1.049,60 €**

Tabla 52: Costes indirectos asociados al proyecto.

Para poder determinar el precio final del proyecto debemos sumar los costes, aplicar el margen de riesgo, el margen de beneficio y los impuestos correspondientes.

Precio final

Total de Coste de horas hombre del proyecto	14.072,50 €
Total de Costes de los Equipos	367,80 €
Total de Costes indirectos	1.049,60 €
COSTES TOTALES	15.489,90 €

Tabla 53: Coste total asociado al proyecto.

Subtotal		15.489,90 €
Margen de riesgo	10%	1.548,99 €
Margen de beneficio	18%	1.703,89 €
Coste del proyecto sin IVA		18.742,78 €
IVA	21%	3.935,98 €
Coste del proyecto IVA incluido		22.678,76 €

Tabla 54: Precio final del proyecto.

El precio final del proyecto es de **veintidós mil seiscientos setenta y ocho euros con setenta y seis céntimos**. Este precio incluye un margen de riesgo de un 10%, sobre él se incluye el 18% de margen de beneficio con lo cual se obtiene el precio sin impuestos, finalmente se suma el 21% de impuesto al valor añadido.

7.3. Entorno socioeconómico

El marco socioeconómico se entiende como el impacto que genera la implantación y el uso de la solución en su entorno. De esta forma, revisaremos cuáles son las principales características y cuáles son los beneficios que obtendremos con el uso de la solución:

- ***Escalabilidad horizontal*:** la solución permite ampliar su capacidad de procesamiento fácilmente y económicamente con la creación de un *cluster*. Esto permite adecuarse a las necesidades de crecimiento de la Universidad.

- **Velocidad de procesamiento:** por medio del *cluster* también es posible mejorar el tiempo de respuesta para consultas de datos muy grandes de forma exponencial. Es importante destacar que una de las previsiones a futuro de este sistema es la gran posibilidad de que el número de ordenadores a monitorizar y el uso de recursos aumenten

significativamente, por lo cual es clave la velocidad de procesamiento para mantener el tiempo de respuesta real.

- **Consultas avanzadas:** el sistema de indexación de ELK, Elasicsearch, permite hacer consultas sobre el uso de recursos con distintos filtros. De esta forma es posible cruzar datos, exportar dichos datos y por medio de una herramienta externa hacer *business intelligence* obtener información depurada que permita identificar problemas que a simple vista son imposibles de detectar.

Gracias a estas características, la herramienta permite:

- Conocer como es el uso de recursos de las aulas en tiempo real
- Mejorar el uso de los recursos por medio del análisis y cruce de datos
- Monitorizar en tiempo real una cantidad mucho mayor de ordenadores, *servidores* y aulas informáticas.

Lo cual se traduce en menores costes, aumentar la capacidad de monitorización de los técnicos de laboratorio y en definitiva **aprovechar mejor el uso de los recursos de la Universidad.**

8. Conclusiones y trabajos futuros

En este último capítulo analizaremos las posibles mejoras del proyecto, las posibilidades de expansión de la usabilidad y las conclusiones finales del mismo. La intención es poder conocer objetivamente las opciones que existen de poder ampliar el alcance del mismo.

8.1. Conclusiones

Las conclusiones son un pequeño resumen de la experiencia que ha aportado este documento. Destacaremos principalmente como ha aportado positivamente a mi futuro profesional, cuál es la experiencia que se ha adquirido en el desarrollo del mismo y en la planificación.

A nivel personal:

Este trabajo ha generado una profunda curiosidad en mi: ¿cómo se tratan grandes cantidades de información en sistemas críticos?, ¿las bases de datos relacionales están preparadas para tratar inserciones y recuperaciones masivas de documentos de *logs* con gran cantidad de términos?, ¿que tan complejo es almacenar y devolver esta cantidad de datos en tiempo real?

Hace dos meses trabajo para el banco Santander y he observado que el almacenamiento de grandes cantidades de *logs* para maquinas que trabajan en el área de producción del banco es completamente inviable. Me pregunte cual era el motivo, llegando a la conclusión de que se debe al mismo que generó la realización de este documento: la gran concurrencia de usuarios y la masiva cantidad de información que produce cada uno de ellos en los *logs*.

Claramente, la sociedad de la información del siglo XXI trata con paradigmas distintos a los que se veían el siglo pasado. Por esto, la postura frente a problemas de este tipo es simple, los sistemas transaccionales **presentan ciertas limitaciones para cantidades de datos muy grandes.**

Es muy probable que próximamente nos veamos enfrentados a soluciones más robustas que permitan recuperar la información que se está desperdiciando, para luego aplicar algoritmos que permitan reconocer modelos o patrones y estos a su vez, permitan explotar más aun el negocio o los recursos de una organización.

Gracias a las asignaturas que he visto en la carrera he podido contar con las herramientas necesarias para comprender esta situación. Asignaturas como “Ficheros y bases de datos” y otras como “Diseño y administración de bases de datos” me han permitido comprender en gran parte el paradigma de las bases de datos y sus limitaciones, para el mundo relacionado con la arquitectura de sistemas *horizontalmente escalables* asignaturas como “Sistemas Distribuidos” me han

permitido entender como los sistemas de información distribuidos se comunican entre sí.

Me gustaría dar las gracias a los todos los profesores de todas las asignaturas vistas en la carrera, a todos aquellos que han puesto un énfasis especial en enseñarnos y en especial a los profesores del grupo Arcos y LaBDA ya que gracias a ellos y a la confianza que pusieron en mi hoy puedo estar escribiendo este trabajo y estar matriculado en uno de los Master que imparte esta Universidad.

Sobre el producto:

Como hemos visto en puntos anteriores, ELK cumple claramente con los objetivos propuestos en la introducción de este documento. Esta herramienta permite explotar mejor los recursos de la Universidad y ampliar los conocimientos que se tienen del uso de recursos en el laboratorio del Departamento de Informática.

De igual forma repasaremos los objetivos que se han cumplido, ya que en sí, este es el propósito principal de este documento:

Objetivos cumplidos:

- ✓ La solución debe ser capaz de realizar las mismas tareas que el actual sistema de monitorización de la Universidad (Munin):
  - ✓ monitorizar el uso de recursos.
  - ✓ mostrar gráficos por la interfaz.
  - ✓ mostrar un histórico de uso de recursos.

- ✓ La solución debe permitir búsquedas de la información contenida en los *logs*:
  - ✓ Es posible obtener la información parcial o total de los mismos.

- ✓ La solución debe permitir la obtención de dicha información por medio de los recursos del laboratorio de informática.

- ✓ La interfaz gráfica de la solución debe permitir hacer *querys* con distintos criterios y detalles.
  - ✓ El resultado debe mostrarse de forma gráfica o poder obtenerse como un conjunto resultado de datos.

- ✓ La solución será preferentemente de licencia gratuita y *open source*.

- ✓ El mantenimiento de la solución debe ser nulo o poco costoso en horas hombre y en coste de equipos (implantación y mantenimiento).

Objetivos no cumplidos:

- Ninguno -

Sobre el proceso del proyecto:

En todo momento la intención fue clara: poder mejorar el sistema previo de monitorización. El proceso desde analizar los requisitos del cliente hasta obtener el precio final del mismo fue largo, pero enriquecedor.

Claramente, las necesidades que se han cubierto con este proyecto nos han permitido visualizar la magnitud del mismo. Mejoras muy pequeñas implican grandes cambios.

Aun así, podemos destacar que el desarrollo del proyecto se llevó a cabo en el tiempo estipulado y no fue necesario el uso del margen de tiempo del 10% que se había previsto al principio. Es necesario destacar que **el tiempo de duración real del proyecto se acerca mucho a lo estipulado por la normativa de horas de los 12 créditos ECTS**, debido a las 2 semanas de cierre por vacaciones de la Universidad.

Al principio, hemos visto que la planificación inicial hacia viable el trabajo, por eso se ha llevado a cabo. Esta planificación inicial se ha estructurado basándonos en los conocimientos adquiridos en la asignatura Dirección de Proyectos de Desarrollo de Software y siguiendo en términos generales la estructura temporal del proyecto hemos podido cumplir con la fecha de entrega.

La clara orientación de la especialidad de Sistemas de Información de la carrera me ha permitido visualizar que éste proyecto era más interesante que otros para mi futuro profesional, por el hecho de que me acercaba un poco más al mundo del Big Data y porque el trabajo se realizaría en un entorno real.

Por otro lado, tenemos que destacar una cosa, la realización de esta práctica podría haberse paralelizado entre los puntos estudio de mercado y análisis de requisitos. Por el hecho de que esta práctica es unipersonal, se han desarrollado los apartados de forma secuencial, pero en caso de que fuese necesario, ambos apartados son independientes y paralelizables, lo cual podría recortar el desarrollo del proyecto en dos semanas manteniendo el coste de horas hombre.

8.2. Trabajos futuros

En este apartado revisaremos las posibles mejoras y ampliaciones de la usabilidad y el alcance del proyecto.

Una posible mejora sería crear una **aplicación para dispositivos móviles que conecte con Kibana para conocer el estado de los *servidores* del laboratorio**, analice ese estado y al encontrar **información anómala envíe la alerta asociada**. De esta forma los técnicos del laboratorio del Departamento de Informática tendrían un tiempo de respuesta más corto para tratar problemas de este tipo.

Como trabajo futuro también se propone recuperar más información que permita entender mejor aquellos problemas derivados del uso de los ordenadores de las aulas, tales como: detectar los instantes en los cuáles las aulas están más ocupadas, determinar qué horas son más apropiadas para mantener los ordenadores en suspensión y ahorrar energía, detectar problemas asociados al desgaste de los ordenadores debido al uso, saber que ordenador y en que aula está libre para que un estudiante pueda trabajar, etc.

Con la información existente y la que se propone recuperar adicionalmente, se podría hacer estudios para ver la correlación entre las mismas. Una herramienta como Microsoft Power BI (26) permitiría hacer consultas a Elasticsearch y extraer posibles relaciones que no sean visibles a simple vista usando *business intelligence*.

Utilizando sistemas de ***Machine Learning***, podríamos aprovechar la información recuperada de los *logs* y obtener modelos predictivos que nos permitan identificar los momentos de pico de uso de recursos a lo largo del año lectivo. Una opción viable y completamente integrable con ELK es Prelert (27).

Por último, integrar *Windows* en la solución también sería una opción como trabajo futuro, esto permitiría recuperar información independiente del sistema operativo que este iniciado en la máquina.

Apéndices

En este apartado incluiremos tres elementos importantes que van dirigidos a aquellas personas que necesiten información más detallada sobre el proyecto. Los tres puntos implicados son:

- **Manual de instalación y configuración:** es una ampliación de lo explicado en el **Capítulo 5** de este documento, con la diferencia de que va dirigido a los técnicos del laboratorio de informática y a todas aquellas personas encargadas de la instalación y configuración del sistema. Este apéndice está orientado a ser un documento anexo que pueda ser impreso para su consulta rápida.

- **Manual de Utilización:** similar al apéndice anterior, con la diferencia de que va orientado al uso del sistema. Explica a grandes rasgos los elementos más útiles de Kibana y sus funciones.

- **Detalle de la planificación, diagrama de Gantt:** planificación del proyecto orientada a ser impresa en horizontal. Permitirá a los integrantes del equipo del proyecto poder visualizar las fases del mismo de forma más detallada.

Es importante agregar que para estos apartados las referencias han sido ingresadas explícitamente, ya que en caso de que sean impresas por separado el lector no podría identificar directamente los enlaces de las mismas.

Apéndice I: Manual de instalación y configuración

Manual de Instalación y configuración de ELK

¡Bienvenido!, este documento le explicará pasa a paso el procedimiento a seguir para la instalación y configuración del sistema ELK en el entorno de las Aulas del laboratorio del Departamento de informática de la Universidad Carlos III de Madrid.

¡Comencemos!

Pre-requisitos

Antes de continuar es necesario identificar cuáles son los requisitos que deben cumplirse previamente a la instalación del sistema. Revisaremos detalladamente cada uno de ellos, indicando el comando necesario para su ejecución cuando corresponda.

La lista de requisitos es la siguiente:

- Instalación del *S.O. Debian* versión "Jessie" 8.5, instalación estándar.
- Configuración de la red *LAN* y *servidores DNS* del laboratorio.
- Instalación del software de *servidor SSH*, instalación por medio de línea de comandos.
    - apt-get install ssh
- Comando sudo, instalación por medio de línea de comandos.
    - apt-get install sudo
- instalación del *servidor* web Apache 2 según el tutorial de ite.educacion.es
    - http://www.ite.educacion.es/formacion/materiales/85/cd/linux/m3/instalacin_y_configuracin_de_apache.html
- Ejecutar apt-get upgrade y apt-get update.
- Creación de una cuenta sin privilegios de súper usuario en el ordenador *nodo* maestro de Elasticsearch.
    - useradd –p elastic nMaestro –g nobody
- Apertura de puertos necesarios para la correcta ejecución del sistema en el *nodo* maestro:
    - Iptables -I INPUT -p all -j DROP
    - iptables -I INPUT -p tcp --dport 23 -j ACCEPT
    - iptables -I INPUT -p tcp --dport 80 -j ACCEPT
    - iptables -I INPUT -p tcp --dport 443 -j ACCEPT
    - iptables -I INPUT -p tcp --dport 5601 -j ACCEPT
- Incluir el *script* de la configuración de *iptables* en el arranque del sistema.
- Instalación de JRE de Java en su versión 8.

Configuración de los equipos

Primero debemos configurar el *nodo* maestro, ya que en él podremos almacenar y visualizar la información recuperada desde las aulas informáticas. En este *nodo* implantaremos dos de las tres capas de la arquitectura: la capa de indexación y la capa de visualización. Esto implica que Kibana y Elasticsearch serán implantados en este ordenador.

La dirección IP de a configurar para el *nodo* maestro es la que ya se ha indicado: **163.117.142.181, esta dirección IP será recuperada automáticamente por medio de *DHCP*.**

El primer paso para tener el *nodo* maestro funcionando es instalar Elasticsearch, para ello dentro de un *terminal* en la línea de comando debemos ejecutar lo siguiente:

```
1 wget -qO - https://packages.elastic.co/GPG-KEY-elasticsearch | sudo apt-key add -
2 echo "deb http://packages.elastic.co/elasticsearch/1.7/debian stable main" | sudo tee –a /etc/apt/sources.list.d/elasticsearch-1.7.list
3 sudo apt-get update
4 sudo apt-get install elasticsearch
```

Con esto agregaremos el repositorio de Elasticsearch para apt para luego instalarlo. El siguiente paso es incluir la siguiente información el fichero de configuración en la ruta: **/etc/elasticsearch/elasticsearch.yaml**

```
1 cluster.name: elasticnode
2 discovery.zen.ping.multicast.enabled: false
3 index.number_of_replicas: 0
```

Luego de esto, procederemos a instalar Kibana, para esto, debemos ingresar los siguientes comandos.

```
1 cd /opt
2 sudo wget https://download.elastic.co/kibana/kibana/kibana-4.1.2-linux-x64.tar.gz
3 sudo tar xvzf kibana-4.1.2-linux-x64.tar.gz
```

Luego, en la ruta **/etc/systemd/system/kibana.service** debemos crear un fichero con la siguiente información dentro.

```
1 [Unit]
2 Description=open source browser based analytics and search dashboard for Elasticsearch
3
4 [Service]
5 Type=simple
6 ExecStart=/opt/kibana-4.1.2-linux-x64/bin/kibana  -c /opt/kibana-4.1.2-linux-x64/config/kibana.yml
```

Después, debemos iniciar los servicios Kibana, Elasticsearch y Apache.

```
sudo systemctl start apache2
sudo systemctl start elasticsearch
sudo systemctl start kibana
```

Con esto ya tenemos el *nodo* maestro funcionando, el siguiente paso es configurar los ordenadores de las aulas para que envíen información al *nodo* maestro. Para esto debemos ejecutar las siguientes líneas de comando en una *terminal* en cada uno de ellos.

```
wget -qO - https://packages.elasticsearch.org/GPG-KEY-elasticsearch | sudo apt-key add -
echo "deb http://packages.elasticsearch.org/logstash/1.5/debian stable main" | sudo tee -a /etc/apt/sources.list
sudo apt-get update
sudo apt-get install logstash
```

Con esto tenemos Logstash instalado, lo siguiente es configurar los filtros para hacer que haga su trabajo, un ejemplo de un filtro es:

```
input {
  file {
    path => '/var/log/apache2/access.log'
  }
}

filter {
  grok {
    match => { "message" => "%{COMBINEDAPACHELOG}" }
  }
}

output {
  stdout { codec => rubydebug }
}
```

Donde el parámetro “path” indica la ruta del *log* a recuperar, el parámetro “match” indica que tipo de *log* se va a recuperar y el output indica que la salida es la estándar. Todos los parámetros son configurables siguiendo la estructura de la configuración de un filtro de Logstash.

La idea principal de los filtros es hacer que Logstash recupere la información de entrada que está señalada, la analice o filtre para luego enviarla a una salida determinada. Cada uno de estos apartados representa las respectivas partes de este proceso.

Luego de crear el filtro, es necesario poner el archivo en la siguiente ruta: **/etc/logstash/conf.d/nombre_del_filtro.conf**, por último, es necesario informar a Logstash de que debe actualizar su lista de filtros.

```
sudo /opt/logstash/bin/logstash -f /etc/logstash/conf.d
```

Para más información sobre la creación de filtros, el lector puede recurrir a la página oficial de Logstash:

https://www.elastic.co/products/logstash.

o a una web de tutorial de creación de *logs*:

https://www.adictosaltrabajo.com/tutoriales/logstash/.

Para seguir un tutorial sobre como configurar ELK, puede recurrir a:

https://www.linode.com/docs/databases/elasticsearch/visualizing-apache-webserver-logs-in-the-elk-stack-on-debian-8.

Además, debemos configurar la página inicial de la aplicación en el *nodo* maestro para habilitar la configuración htaccess de Apache y evitar el acceso no autorizado. Para esto primero, cambiaremos la información de la página de inicio de Apache para que redirija automáticamente a la página de Kibana siguiendo este ejemplo:

```
  GNU nano 2.2.6                Fichero: index.html

<!DOCTYPE HTML>
<html>
  <head>
    <meta http-equiv="refresh" content="0; url=http://163.117.142.181:5601/" />
  </head>
  <body>
  </body>
</html>
```

Este fichero no contiene información, pero por medio del comando “refresh” re direcciona automáticamente a la *URL* especificada en el campo “url”. Lo siguiente es configurar un fichero .htpasswd y un fichero .htaccess. Para el primer fichero debemos crear ambos ficheros en blanco con el siguiente comando:

```
1  touch /var/www/html/.htpasswd
2  touch /var/www/html/.htaccess
```

La recuperación de estos ficheros por medio de http desde el exterior esta desactivada por defecto en Apache desde la versión 2, lo cual nos permite almacenar las contraseñas de forma segura. Luego, debemos editar el fichero .htaccess agregando la siguiente información

```
1  AuthName "Se necesita un nombre de usuario y una contraseña "
2  AuthType Basic
3  require valid-user
4  AuthUserFile /var/www/html
```

Finalmente, debemos crear un usuario para que pueda iniciar sesión en el sistema, los datos para ello serán: **Usuario:** “kibana”, **Contraseña:** “articuno”

El comando para configurar los datos de acceso es:

```
1 htpasswd /var/www/html kibana
```

Apache solicitara ingresar la nueva contraseña para el usuario Kibana y confirmarla, es necesario escribir la contraseña en la *terminal* exactamente igual ambas veces. Finalmente, el último paso es generar el alias del fichero htaccess para hacer que la configuración funcione, para esto debemos editar el fichero **default.conf** dentro de la ruta **/etc/apache2/sites-avalibles/** agregando la información que se muestra en la siguiente imagen.

```
 GNU nano 2.2.6                          Fichero: /etc/apache2/sites-available/000-default.conf

<VirtualHost *:80>
        # The ServerName directive sets the request scheme, hostname and port that
        # the server uses to identify itself. This is used when creating
        # redirection URLs. In the context of virtual hosts, the ServerName
        # specifies what hostname must appear in the request's Host: header to
        # match this virtual host. For the default virtual host (this file) this
        # value is not decisive as it is used as a last resort host regardless.
        # However, you must set it for any further virtual host explicitly.
        #ServerName www.example.com

        ServerAdmin webmaster@localhost
        DocumentRoot /var/www/html

        Alias /html /var/www/html
        <Directory "/var/www/html">
          Options Includes
          AllowOverride All
          Order allow,deny
          Allow from all
        </Directory>


        # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
        # error, crit, alert, emerg.
        # It is also possible to configure the loglevel for particular
        # modules, e.g.
        #LogLevel info ssl:warn

        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log combined

        # For most configuration files from conf-available/, which are
        # enabled or disabled at a global level, it is possible to
        # include a line for only one particular virtual host. For example the
        # following line enables the CGI configuration for this host only
        # after it has been globally disabled with "a2disconf".
        #Include conf-available/serve-cgi-bin.conf
</VirtualHost>
```

Los apartados que debemos agregar son la línea Alias y las seis líneas siguientes. Para más información de cómo configurar Apache para el uso de htaccess se recomienda al lector revisar el siguiente tutorial:

http://www.alcancelibre.org/staticpages/index.php/18-como-apache-htaccess.

La ultima configuración necesaria de Apache es la inclusión del protocolo SSL para la conexión segura con el sitio *web*. La configuración estándar de Apache se puede revisar en el tutorial de la *web* de digicert.com (25). Los pasos para configurar el certificado SSL auto firmado son los siguientes:

```
openssl genrsa -out ca.key 1024
openssl req -new -key ca.key -out ca.csr
openssl x509 -req -days 365 -in ca.csr -signkey ca.key -out ca.crt
```

Luego de crear y firmar los certificados, es necesario moverlos a una ruta donde puedan ser accedidos por Apache, para ello:

```
mv ca.crt /etc/pki/tls/certs
mv ca.key /etc/pki/tls/private/ca.key
mv ca.csr /etc/pki/tls/private/ca.csr
```

Después de eso, debemos forzar a que Apache use los certificados, modificando las líneas del fichero **/etc/httpd/conf/ssl.conf**

```
SSLCertificateFile /etc/pki/tls/certs/ca.crt
SSLCertificateKeyFile /etc/pki/tls/private/ca.key
```

Para luego configurar el fichero que permite la conexión segura, este fichero es el siguiente: **/etc/httpd/conf/https.conf**, y las líneas a agregar son:

```
NameVirtualHost *:443

<VirtualHost *:443>
    SSLEngine on
    SSLCertificateFile /etc/pki/tls/certs/ca.crt
    SSLCertificateKeyFile /etc/pki/tls/private/ca.key

    AllowOverride All

    DocumentRoot /var/www/httpsdocs
    ServerName nombresitio.com
</VirtualHost>
```

Con esto último creamos un nuevo huésped virtual, que nos permite iniciar comunicaciones seguras. Para asegurarnos de que la configuración funciona perfectamente debemos reiniciar el *daemon* de Apache:

```
sudo systemctl restart apache2
```

Y en los ordenadores de las aulas debemos reiniciar el *daemon* de Logstash con el siguiente comando:

```
sudo systemctl restart logstash
```

Haciendo esto habremos finalizado la configuración del sistema ELK implantado dentro de las aulas informáticas del laboratorio del Departamento de Informática. La configuración restante es en sí misma la personalización que los técnicos del laboratorio deseen ejecutar, entiéndase el *dashboard* de Kibana y los filtros necesarios para la recuperación de los *logs* que estimen convenientes.

¡ENHORABUENA!, buen trabajo =)

FIN DEL DOCUMENTO.

Apéndice II: Manual de Utilización

Manual Utilización de Kibana

¡Bienvenido!, este documento le explicará pasa a paso como usar Kibana, parte de la Suite ELK en el entorno de las Aulas del laboratorio del Departamento de informática de la Universidad Carlos III de Madrid.

¡Comencemos!

Acceso y Log-in

Antes de poder acceder a Kibana es necesario iniciar sesión, para ello debemos ejecutar los siguientes pasos:

1) Debemos abrir un *navegador web* compatible con HTML 5, por ejemplo, Firefox
2) Debemos introducir la siguiente dirección web en el navegador

http://163.117.142.181/

3) El sistema nos indicara que debemos introducir el usuario y contraseña proporcionados por el laboratorio del Departamento de Informática de la Universidad, luego, debemos hacer *click* en el botón “iniciar sesión”.

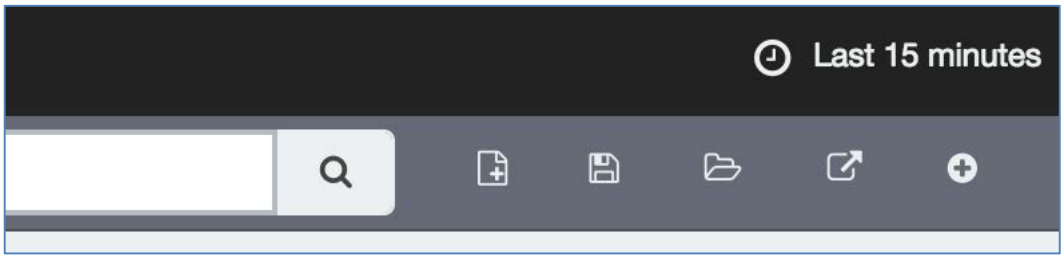
Revisión de las opciones de Kibana

Luego se haber hecho los pasos anteriores tendremos acceso al sistema, específicamente a Kibana, en el cual podremos acceder a las siguientes opciones:

- **Discover**: Pantalla que nos permite filtrar y buscar registros en un intervalo de tiempo determinado.
- **Visualize**: Pantalla donde se pueden crear, ajustar, modificar y ver visualizaciones personalizadas del sistema (Gráficos, tablas, etc.).
- **Dashboard**: Pantalla donde se pueden crear, ajustar, modificar y ver sus propias visualizaciones de forma personalizada.
- **Settings**: Pantalla que permite cambiar la configuración por defecto del sistema o también patrones de índice de Elasticsearch.

Por medio de estas opciones podemos acceder a la gran mayoría de las actividades que nos permite ejecutar el sistema. Aquellas opciones que no son configurables desde Kibana corresponden a las opciones de implantación y configuración del sistema, por lo cual, el acceso a ellas debería ser autorizado por los técnicos del laboratorio del Departamento de Informática de la Universidad.

Las opciones del menú de archivo de Kibana que son referenciadas en este manual corresponden a las que se pueden ver en esta imagen:



La opción superior, donde aparece el reloj indica la configuración del tiempo de refresco del gráfico. Las cinco opciones de la parte inferior corresponden a: **New** (crear nuevo elemento), **Save** (almacenar este elemento), **Load** (cargar un elemento guardado), **Share** (compartir el elemento en las redes sociales) y **Options** (en el caso de Visualize es refrescar y en el caso de *Dashboard* es agregar un nuevo grafico).

La barra de búsqueda permite realizar *query*s al sistema de indexado del sistema.

Configuración inicial de Kibana

El primer paso antes de empezar es configurar un índice. Un índice es un destino en el cual Kibana puede acceder a recuperar información indexada, es decir, estamos intentando enlazar Kibana y Elasticsearch. Para agregar un nuevo índice los pasos son:

1) Acceder a la pestaña **Settings**, acceder a la opción **Configure an index pattern**, configurar las opciones del índice y hacer *click* en el botón **Create.** Si esta es la primera vez que se crea el índice, las opciones deben dejarse por defecto, de esa forma Kibana y Elasticsearch se comunican automáticamente.

2) Volver a la pestaña **Discover** y proceder a la recuperación de datos gráficos.

Luego, en la pestaña **Discover** podemos agregar gráficos y vistas que nos permitan monitorizar los recursos de la universidad, tales como: ordenadores, aulas y *servidores* que estén generando información, para esto, previamente debe haberse implantado el componente Logstash del sistema en dichos equipos.

Creación de un Gráfico de monitorización

Una vez estos equipos generen información, Kibana será capaz de poder mostrarla en los gráficos. Para proceder a crear un gráfico, es necesario seguir los siguientes pasos:

1) Hacer *click* en el botón **New Visualization**, seleccionar el tipo de gráfico que deseamos visualizar y seleccionar una búsqueda nueva o una guardada.
2) Visualizar el contenido del gráfico, verificar si es lo que buscamos obtener, si es así, almacenamos el gráfico haciendo *click* en el botón **Save Visualization.**
3) Repetir el proceso las veces que sea necesario para obtener todos los gráficos que necesitamos incluir en el *dashboard*.

Luego de haber hecho los pasos anteriores ya podemos monitorizar los recursos de la universidad. El siguiente paso es componer un **Dashboard**, en el podremos poner juntos distintos gráficos que se ajusten a nuestras necesidades.

Creación de un Dashboard

En el **Dashboard** podremos poner un conjunto de gráficos en función de lo que necesitemos. Esta opción de Kibana nos permite añadir y eliminar gráficos existentes, el conjunto de estos gráficos nos permite analizar distintas situaciones relacionadas con los recursos de la Universidad desde un navegador conectado a Kibana.

Para crear un ***dashboard*** debemos seguir los siguientes pasos:

1) Acceder a la pestaña **Dashboard**, hacer *click* en el botón **New Dashboard** y proceder a agregar los gráficos almacenados que necesitemos.
2) Configurar los tamaños y ubicaciones de los gráficos según sea necesario, esta configuración puede editarse en caso de que el grafico deba ser modificado más adelante.
3) Proceder a guardar el grafico, para esto debemos hacer *click* en el botón **Save Dashboard.**

Recuperación de información avanzada del sistema

La última opción que debemos repasar del sistema es la creación de *querys* que permitan seleccionar nueva información. Para acceder a esto debemos hacer *click* en la opción **Discover** de Kibana.

Los pasos necesarios para poder acceder a la barra de búsqueda son:

1) Hacer *click* en la barra blanca alargada situada en la parte superior de la interfaz de Kibana.
2) Ingresar una consulta basándonos en los entandares de consultas de Elasticsearch, el manual de cómo poder ejecutar una *query* puede encontrarse directamente en la web de Elastic en el siguiente enlace.

https://www.elastic.co/guide/en/kibana/3.0/queries.html

3) Hacer *click* en el icono de la lupa situado a la derecha de la barra, esperar y revisar los resultados obtenidos de la consulta.
4) En caso de ser necesario, almacenar la búsqueda haciendo *click* en el botón **Save Search**, con ello podremos usar la vista para obtener nuevos gráficos.

Siguiendo estas instrucciones podremos usar correctamente todas las opciones de Kibana, podremos configurar el *dashboard* del laboratorio de forma óptima y con ello podremos trabajar de forma más rápida y cómoda.

¡ENHORABUENA!, buen trabajo =)

FIN DEL DOCUMENTO.

Apéndice III: Detalle de la planificación, diagrama de Gantt.

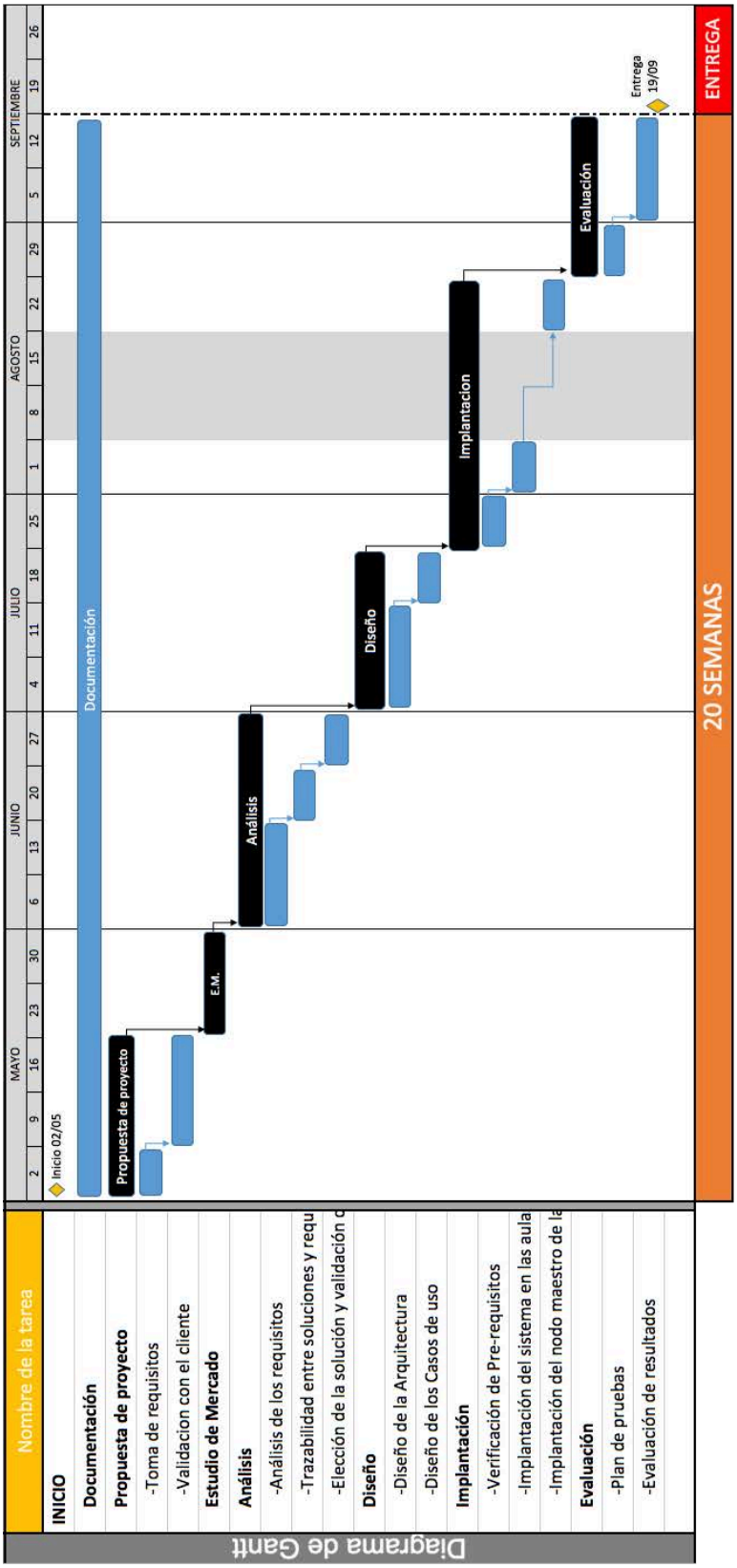


Ilustración 31: Carta Gantt de la planificación del proyecto puesta de forma horizontal para su correcta visualización.

Bibliografía

1. **munin.** www.munin-monitoring.org. *www.munin-monitoring.org.* [En línea] 05 de 09 de 2016. www.munin-monitoring.org.

2. **munin-monitoring.org.** http://munin-monitoring.org/ . *http://munin-monitoring.org/* . [En línea] [Citado el: 26 de 09 de 2016.] http://munin-monitoring.org/ .

3. —. http://guide.munin-monitoring.org/en/latest/. *http://guide.munin-monitoring.org/en/latest/.* [En línea] [Citado el: 26 de 09 de 2016.] http://guide.munin-monitoring.org/en/latest/.

4. **nagios.org.** https://www.nagios.org/. *https://www.nagios.org/.* [En línea] [Citado el: 26 de 09 de 2016.] https://www.nagios.org/.

5. —. https://www.nagios.org/projects/nagios-core/. *https://www.nagios.org/projects/nagios-core/.* [En línea] https://www.nagios.org/projects/nagios-core/.

6. —. https://www.nagios.org/support/. *https://www.nagios.org/support/.* [En línea] 26 de 09 de 2016. https://www.nagios.org/support/.

7. **nagios-cl.org.** http://www.nagios-cl.org/que-es-nagios. *http://www.nagios-cl.org/que-es-nagios.* [En línea] 26 de 09 de 2016. http://www.nagios-cl.org/que-es-nagios.

8. **nagios.com.** https://www.nagios.com/products/nagios-xi/. *https://www.nagios.com/products/nagios-xi/.* [En línea] [Citado el: 26 de 09 de 2016.] https://www.nagios.com/products/nagios-xi/.

9. **ganglia.info.** http://ganglia.info/. *http://ganglia.info/.* [En línea] [Citado el: 26 de 09 de 2016.] http://ganglia.info/.

10. **github.com.** https://github.com/. *https://github.com/.* [En línea] https://github.com/.

11. **ganglia.info.** http://ganglia.info/?page_id=68. *http://ganglia.info/?page_id=68.* [En línea] [Citado el: 26 de 09 de 2016.] http://ganglia.info/?page_id=68.

12. **splunk.com.** http://www.splunk.com/. *http://www.splunk.com/en_us/solutions/solution-areas/it-operations-management/server-and-network-infrastructure-monitoring.html. .* [En línea] [Citado el: 26 de 09 de 2016.] http://www.splunk.com/en_us/solutions/solution-areas/it-operations-management/server-and-network-infrastructure-monitoring.html. .

13. **elastic.co.** https://www.elastic.co/. *https://www.elastic.co/webinars/introduction-elk-stack.* [En línea] [Citado el: 26 de 09 de 2016.] https://www.elastic.co/webinars/introduction-elk-stack.

14. **esa.int.** http://www.esa.int/ESA. *http://www.esa.int/ESA.* [En línea] http://www.esa.int/ESA.

15. **administracionelectronica.gob.es.** http://administracionelectronica.gob.es. *http://administracionelectronica.gob.es/pae_Home/pae_Documentacion/pae_Metodolog/pae*

_Metrica_v3.html. [En línea] [Citado el: 26 de 09 de 2016.] http://administracionelectronica.gob.es/pae_Home/pae_Documentacion/pae_Metodolog/pae_Metrica_v3.html.

16. **iso.org.** http://www.iso.org/. *http://www.iso.org/iso/catalogue_detail?csnumber=43447.* [En línea] 26 de 09 de 2016. http://www.iso.org/iso/catalogue_detail?csnumber=43447.

17. **aenor.es.** http://www.aenor.es/. *http://www.aenor.es/aenor/certificacion/calidad/calidad_software_15504.asp#.V-lqZZOLSAx.* [En línea] [Citado el: 26 de 09 de 2016.] http://www.aenor.es/aenor/certificacion/calidad/calidad_software_15504.asp#.V-lqZZOLSAx.

18. **elastic.co.** https://www.elastic.co/. *https://www.elastic.co/guide/en/logstash/current/filter-plugins.html.* [En línea] [Citado el: 26 de 09 de 2016.] https://www.elastic.co/guide/en/logstash/current/filter-plugins.html.

19. **blog.desdelinux.net.** http://blog.desdelinux.net/. *http://blog.desdelinux.net/instalacion-configuracion-debian-8-gnome/.* [En línea] [Citado el: 26 de 09 de 2016.] http://blog.desdelinux.net/instalacion-configuracion-debian-8-gnome/.

20. **ite.educacion.es.** ite.educacion.es. *ite.educacion.es.* [En línea] 19 de 9 de 2016. http://www.ite.educacion.es/formacion/materiales/85/cd/linux/m3/instalacin_y_configuracin_de_apache.html.

21. **www.elastic.co.** www.elastic.co. *www.elastic.co.* [En línea] 19 de 09 de 2016. https://www.elastic.co/products/logstash.

22. **adictosaltrabajo.com.** www.adictosaltrabajo.com. *www.adictosaltrabajo.com.* [En línea] 19 de 09 de 2016. https://www.adictosaltrabajo.com/tutoriales/logstash/.

23. **www.linode.com.** www.linode.com. *www.linode.com.* [En línea] 19 de 09 de 2016. https://www.linode.com/docs/databases/elasticsearch/visualizing-apache-webserver-logs-in-the-elk-stack-on-debian-8.

24. **alcancelibre.org.** alcancelibre.org. *alcancelibre.org.* [En línea] 19 de 9 de 2016. http://www.alcancelibre.org/staticpages/index.php/18-como-apache-htaccess.

25. **www.digicert.com.** https://www.digicert.com/. *https://www.digicert.com/.* [En línea] [Citado el: 26 de 09 de 2016.] https://www.digicert.com/es/instalar-certificado-ssl-apache.htm.

26. **microsoft.com.** http://powerbi.microsoft.com. *http://powerbi.microsoft.com.* [En línea] https://powerbi.microsoft.com/es-es/.

27. **http://prelert.com.** http://prelert.com. *http://prelert.com.* [En línea] [Citado el: 26 de 09 de 2016.] http://info.prelert.com/products/behavioral-analytics-for-the-elastic-stack).

28. **https://www.elastic.co/guide/en/logstash/current/filter-plugins.html. *https://www.elastic.co/guide/en/logstash/current/filter-plugins.html.* [En línea] https://www.elastic.co/guide/en/logstash/current/filter-plugins.html.**

- Fin del documento -